

Министерство образования и науки Российской Федерации
ФГБОУ ВО «Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра физики и математического моделирования

«Разработка системы автоматизации помещения, с использованием облачных технологий»

Выпускная квалификационная работа
Направление: 09.03.03 – прикладная информатика
Профиль: Прикладная информатика в экономике

Квалификационная работа
допущена к защите
Заведующий кафедрой
д.ф.-м. н., профессор
Сидоров Валерий Евгеньевич

дата

подпись

Исполнитель:
Нечаев Николай
Александрович
обучающийся БЭ-51z группы

подпись

Руководитель:
Омельченко Сергей Владимирович
к.т.н., доцент

подпись

Екатеринбург 2017

Содержание

Введение.....	3
Глава 1. Автоматизированные системы и облачные технологии, теоретические основы.....	5
1.1. Автоматизированные системы.....	5
1.1.1. Общие сведения.....	5
1.1.2. Применение и основные функции АСУ.....	7
1.1.3. Информационная база АСУ.....	8
1.1.4. Техническая база АСУ.....	9
1.1.5. Механические средства сбора и вывода информации.....	10
1.2. Микроконтроллеры, общий обзор.....	11
1.2.1. Микроконтроллеры на ядре AVR.....	13
1.3. Облачные технологии, краткая история.....	17
1.3.1. Облачные сервисы и их типы.....	19
1.3.2. Модели обслуживания, предоставляемые облачными сервисами.....	20
1.3.3. Модели развертывания облачных сервисов.....	21
Глава 2. Разработка системы контроля микроклимата в офисном помещении.....	23
2.1. Архитектура аппаратных средств.....	23
2.2. Программно-аппаратные средства для системы контроля микроклимата.....	25
2.2.1 Микроконтроллер.....	25
2.2.2. Используемые датчики.....	28
2.3. Программные средства, для системы контроля микроклимата в офисном помещении.....	33
2.3.1. Среда программирования Arduino IDE.....	33
2.3.2. Система домашней автоматизации MajorDoMo.....	39
2.3.3. Сервис облачной синхронизации и управления CloudSync.....	45
2.3.4. Программирование микроконтроллера.....	47
Заключение.....	49
Список литературы.....	50
Приложение 1.....	53
Приложение 2.....	54

Введение

Понятие «автоматизированная система управления» в России начали применять в 50-е годы XX века. Активное использование этих систем началось в 1970-е годы. Оно было нацелено на освобождение человека от выполнения однообразных или опасных видов работ. Возникновение автоматизированных систем управления обосновано потребностью в совершенствовании организационной структуры управления предприятием, организацией, учреждением и т. д. Сейчас это необходимый компонент многих отраслей производства. Автоматизированные системы управления, это комплекс экономических и математических методов, программных и аппаратных средств, а также персонала, предназначенный для управления разнообразными процессами в рамках заданных целей и в отличие от автоматической системы сохраняет за человеком выполнение некоторых функций.

Автоматизация основывается на широком применении средств электронно-вычислительной техники и разнообразного программного обеспечения. С развитием сети Интернет, расширилась область применения автоматизированных систем управления. В том числе, этому способствует распространенность облачных технологий, при взаимодействии с которыми, стало возможно получать информацию в режиме реального времени из любой точки мира, где возможно подключиться к сети Интернет. Использование автоматизированных систем в совокупности с облачными технологиями позволяет значительно сократить затраты на оборудование и персонал компании, и значительно увеличить скорость доступа к информации.

Актуальность выбранной темы выпускной квалификационной работы очевидна, так как развитие информационных технологий возрастает с каждым годом и внедрение автоматизированных систем управления в объединении с облачными технологиями в повседневную жизнь не является

невозможным. Что доказывает доступность и распространенность систем управления помещениями, например, систем типа «умный дом».

Целью выпускной квалификационной работы является разработка и внедрение на предприятии системы контроля микроклимата помещения.

Для достижения поставленной цели, требуется решить следующие задачи:

1. Изучить теоретические основы по заданной теме;
2. Выбрать программное и аппаратное обеспечение для реализации системы;
3. Объединить аппаратные и программные средства в единую автоматизированную систему;
4. Протестировать и реализовать рабочую модель системы контроля микроклимата.

Объектом исследования выступают автоматизированные системы управления.

Предметом исследования выступает система мониторинга микроклимата помещения.

Выпускная квалификационная работа состоит из введения, двух глав и заключения.

В первой главе рассматриваются теоретические основы систем автоматизации, облачных технологий и микроконтроллеров.

Во второй главе рассмотрен процесс разработки и реализации системы мониторинга микроклимата офисного помещения на практике.

Глава 1. Автоматизированные системы и облачные технологии, теоретические основы.

1.1. Автоматизированные системы.

1.1.1. Общие сведения.

Необходимость увеличивать производительность и эффективность труда специалистов, производить больше качественной продукции привели к созданию автоматизированных систем. Изначально создавались всевозможные автоматические устройства, призванные избавить человека от выполнения разнообразных однообразных и опасных типов работ, например роботы-автоматы и др.

Как правило, под «автоматической» понимают любую саморегулирующуюся систему, принцип действия которой заключается в сопоставлении определенных выходных характеристик с установленным эталоном. Отклонение выходного значения от образца вызывает активацию элементов обратной связи для корректировки полученных данных. Созданные таким образом устройства облегчили, ускорили и, как правило, удешевили выполнение определенных видов работ. В качестве дальнейшего развития таких систем, были разработаны автоматизированные системы [20, 30].

Автоматизированная система – это система, включающая в себя как персонал, так и комплекс электронно-вычислительных средств, в которой часть функций управления выполняет человек. Данная система представляет собой совокупность технических и программных средств, а также персонала, служащую для автоматизации различных процессов. В основном подобные системы разрабатывались в промышленности и предназначались для совершенствования методов управления производственными процессами. Это автоматизированные системы управления и автоматизированные системы управления технологическими процессами [14, 15].

Автоматизированная система управления – это комплекс экономических и математических методов, программных и аппаратных средств, а также персонала, предназначенный для управления разнообразными процессами в рамках заданных целей. В отличие от автоматической системы сохраняет за человеком выполнение некоторых функций, например, не поддающихся автоматизации.

Основное назначение автоматизированной системы управления – повышение эффективности управления объектом на основе роста производительности труда и совершенствования методов планирования процесса управления.

Появление автоматизированных систем управления вызвано потребностью совершенствования организационной структуры управления предприятием, организацией, учреждением и т.п. Автоматизированная система управления представляет собой совокупность коллектива людей и комплекса программно-технических средств, т.е. является человеко-машинной системой, базирующейся на экономико-математических методах управления, использовании средств электронно-вычислительной техники.

В структуре любой автоматизированной системы управления можно выделить следующие компоненты:

Основная часть – включает в себя математическое и информационное обеспечение и техническую часть.

Функциональная часть – подразумевает конкретные управленческие функции и ряд взаимосвязанных программ.

Системы могут быть простыми или объемными и сложными. Различают две разновидности систем – автоматизированная система управления техническим процессом (АСУТП) и система организационного управления (АСОУ). Эти системы различаются между собой характеристиками объектов, которыми управляет система. АСУТП разрабатываются для управления сложными техническими объектами, механизмами, аппаратами, машинами. АСОУ призваны контролировать

функционирование коллективов людей. Согласно применению АСУ, отличаются и способы передачи данных – это как документы, так и различные физические сигналы [11, 16, 34].

1.1.2. Применение и основные функции АСУ.

АСУ применяются в различных областях производства. Основными функциями систем можно назвать:

Автоматизация управления технологическим процессом – в результате использования системы автоматизации происходит оптимизация взаимодействия всех компонентов производства. Экономятся ресурсы, улучшаются показатели процесса.

Сбор, хранение, обработка и выдача информации о состоянии оборудования и процесса, в общем. Информация считывается с датчиков и выводится постоянно или по требованию.

Распознавание и фиксация внештатных ситуаций и отклонений от заданного процесса. В случае возникновения внештатной ситуации система оповещает оператора либо устраняет неполадки в автоматическом режиме, для предотвращения развития аварии.

Предоставление необходимой информации в виде графических и числовых данных. Данные могут выводиться на экран монитора в виде текстовой или графической информации.

Регистрация выполняемых действий и их хранение в архивах. Данные могут иметь временную привязку, и при необходимости, есть возможность выявить и проанализировать причину аварийной ситуации.

Многоуровневая защита информации. Доступ к данным автоматизированной системы, как правило, ограничен и предоставляется только авторизованным пользователям [5, 6, 14].

1.1.3. Информационная база АСУ.

Информационной базой АСУ можно назвать весь комплекс информации, хранящейся на носителях и необходимых для штатной работы системы. Обычно, вся информационная база условно делится на три сектора – генеральный, производный и оперативный.

Генеральный сектор содержит все данные, являющиеся общими для всех поставленных задач. Размещение этих данных не предназначено для выполнения какой-нибудь одной управленческой функции. Если объект довольно большой, генеральный сектор может хранить большие объёмы данных и занимать много места на запоминающих устройствах. Особую сложность в таких случаях может вызывать мультипрограммная обработка совместно с недостаточной мощностью технических средств. В генеральном секторе в обязательном порядке должны отображаться все устойчивые изменения в работе системы.

Производный сектор призван решить указанную выше задачу. Он отображает специфику определенного объекта, особенности функций, выполняющихся в любой определенный отрезок времени и ряд иных показателей системы. Каждый производный сектор формируется из генерального сектора.

Оперативный сектор предназначен для обработки текущей информации и регистрации промежуточных результатов. Здесь же можно найти изначально заданную информацию об обслуживаемом объекте. Эти данные поступают по каналам связи или же находятся на различных носителях данных. Затем данные могут передаваться в производный и генеральный секторы [7, 11, 14].

1.1.4. Техническая база АСУ.

Под технической базой АСУ принято понимать все технические средства, которые применяют для сбора, хранения и обработки информации, а также для её отображения и передачи. Так же сюда могут относиться и исполнительные устройства, воздействующие на объект управления.

Основные технические элементы и оборудование АСУ – это электронно-вычислительная техника, обеспечивающая хранение и обработку данных, используемых системой. Такая техника дает возможность моделировать производственные процессы и строить предложения для управления.

Для создания и управления АСУ используются два типа электронно-вычислительной техники – информационно-расчётный и учётно-регулирующий.

Информационно-расчётное оборудование находится на высшей ступени в системе управления. Данное оборудование занимается решением всех задач, связанных с централизованным управлением объектом. Для таких механизмов свойственно высокое быстродействие, наличие системы прерываний, переменная длина слова, слоговая обработка вводных данных.

Нижнюю ступень системы управления, обычно занимают учётно-регулирующие механизмы и оборудование. Эти устройства устанавливаются непосредственно на участках или в цехах на производстве. Они предназначены для сбора вводных данных от объектов управления, первичной обработки информации, ее передача в информационно-расчётный блок и получение управляющей информации. Также, учётно-регулирующее оборудование выполняет локальные расчёты и вырабатывает управляющие воздействия на объекты управления, в случае появления несоответствий расчётным функциям. Эта часть системы управления характеризуется хорошо развитой связью с широким перечнем источников информации и устройств управления [7, 11, 14, 16].

1.1.5. Механические средства сбора и вывода информации.

Если система подразумевает сбор и обработку информации с привлечением человека, в неё добавляются всевозможные регистраторы, которые дают возможность получать данные напрямую с объектов. Например, сюда можно отнести различные датчики, таймеры, счетчики произведённых деталей и прочее аналогичное оборудование. Также устанавливаются автоматические регистраторы отклонений в производственном процессе, которые фиксируют и передают в систему управления информацию об отсутствии материалов, инструментария, а также ошибки работе оборудования.

К средствам вывода данных относятся все устройства, позволяющие вывести информацию в удобном для человека виде. Например, различные мониторы, печатающие устройства, терминалы, индикаторы и пр. Эти устройства связаны с центральным процессором вычислительной машины и способны предоставлять информацию в реальном времени, через заданные промежутки времени, либо по запросу пользователя или при возникновении внештатной ситуации.

В состав технической базы автоматизированных систем управления также входят различные виды оргтехники, контрольно-измерительные и учётные приборы, обеспечивающие штатное функционирование основных технических узлов [11, 14, 15, 16]. Структура современной промышленной системы автоматизированного управления приведена на рисунке 1.



Рис. 1. Структура современной промышленной системы автоматизированного управления производством.

1.2. Микроконтроллеры, общий обзор.

Микроконтроллер – компьютер на одной микросхеме. Служит для управления разнообразными электронными устройствами и осуществляет взаимосвязь между ними согласно заложенной в микроконтроллер программой. В отличие от микропроцессоров, которые используются в персональных компьютерах, микроконтроллер содержит интегрированные вспомогательные устройства. Эти устройства работают под управлением микропроцессорного ядра микроконтроллера. Контроллером обычно называют плату, созданную на базе микроконтроллера и часто при использовании понятия "микроконтроллер" для простоты употребляют

сокращенное название этого устройства, без приставки "микро". Нередко при упоминании микроконтроллера встречаются слова "чип" или "микрочип", "кристалл" (большинство микроконтроллеров изготавливаются на одном кристалле кремния), сокращения МК или от английского microcontroller – МС [3, 4, 7].

Основным признаком классификации микроконтроллеров является разрядность данных, обрабатываемых арифметико-логическим устройством (АЛУ). Микроконтроллеры подразделяются на 4-, 8-, 16-, 32- и 64-разрядные. На данный момент наибольшая доля мирового рынка микроконтроллеров занята восьмиразрядными устройствами (приблизительно 50 % в стоимостном выражении). Следующими по распространенности идут 16-разрядные и DSP-микроконтроллеры (DSP – Digital Signal Processor – цифровой сигнальный процессор), используемые в системах обработки сигналов (около 20 % рынка на каждую из групп). Каждая группа микроконтроллеров делится на CISC- и RISC-устройства. Наиболее распространенной группой устройств являются CISC-микроконтроллеры, но в последние годы в индустрии производства микрочипов отмечается заметный рост доли RISC-архитектуры. Тактовая частота определяет, количество вычислений выполняемых за единицу времени. Как правило, производительность микроконтроллера, а также потребляемая им мощность возрастают при повышении тактовой частоты. Производительность микроконтроллера измеряют в MIPS (million instructions per second – миллион инструкций в секунду) [3, 4, 7, 12].

В 1996 году произошел скачок в индустрии производства микроконтроллеров, корпорация Atmel представила свою линейку чипов на новом перспективном ядре AVR. Более продуманная архитектура AVR и быстроедействие значительно превосходили контроллеры от Microchip. Привлекательная цена на новое решение склонила многих разработчиков к выбору контроллеров на новом ядре. Система команд микроконтроллера AVR более развита, по сравнению с конкурентами, и насчитывает до 133

инструкций. Производительность, контроллера AVR приближается к 1 MIPS/МГц. Чип оборудован постоянным запоминающим устройством для хранения программ с возможностью внутрисхемного программирования. AVR-архитектура адаптирована для использования языка высокого уровня Си. Также, все чипы семейства обратно совместимы. Большим достоинством является доступность программного обеспечения и средств разработки, компания Atmel производит широкий спектр бесплатных программных продуктов. Не менее важное значение, чем сами кристаллы, имеют развитые средства поддержки разработок при освоении и ознакомлении с любой линейкой микроконтроллеров. Производитель уделяет этому аспекту большое внимание, например, удачная и бесплатная среда разработки AVR Studio, для операционной системы Windows. Сторонние разработчики также производят широкий спектр компиляторов, программаторов, ассемблеров, отладчиков, разъемов и адаптеров. Для начинающего разработчика важным является то, что программировать контроллер на базе AVR возможно без аппаратного программатора. Распространен способ программирования этих микроконтроллеров посредством подключения к параллельному порту персонального компьютера. Можно сказать, что AVR становится еще одним стандартом среди 8-разрядных микроконтроллеров широкого назначения. Они доступны в России и характеризуются невысокой стоимостью, успешно конкурируя с изделиями компании Microchip. Все вышесказанное представляет микроконтроллеры Atmel AVR как одни из самых привлекательных для обучения [2, 8, 9, 10].

1.2.1. Микроконтроллеры на ядре AVR.

Микроконтроллер AVR содержит: быстрый RISC-процессор, два типа энергонезависимой памяти: Flash-память программ и память данных EEPROM (Electrically Erasable Programmable Read-Only Memory) – электрически стираемое перепрограммируемое ПЗУ (ЭСППЗУ),

оперативную память RAM, порты ввода/вывода и разнообразные периферийные интерфейсные схемы.

Основой микроконтроллеров AVR служит 8-битное микропроцессорное ядро или центральное процессорное устройство (ЦПУ), созданное по принципам RISC-архитектуры. Основанием этого блока является арифметико-логическое устройство (АЛУ). По системному тактовому сигналу из памяти программ в соответствии с содержимым счетчика команд (Program Counter – PC) выбирается очередная команда и выполняется АЛУ. При выборе команды из памяти программ выполняется предыдущая выбранная команда, что и способствует достижению быстродействия 1 MIPS на 1 МГц. АЛУ подключено к регистрам общего назначения РОН (General Purpose Registers – GPR). Микропроцессор имеет 32 регистра общего назначения, они имеют байтовый формат – каждый из них состоит из восьми бит. РОН располагаются в начале адресного пространства оперативной памяти, но не являются ее частью. Исходя из этого, существует два способа обращения к ним, как к регистрам и как к памяти. Данное решение это специфика AVR и позволяет увеличить эффективность работы и производительность микроконтроллера. Отличие между регистрами и оперативной памятью состоит в том, что с регистрами можно выполнять любые операции (арифметические, логические, битовые), а в оперативную память возможно только записывать данные из регистров [2, 8, 9, 17, 18].

Микроконтроллеры AVR используют Гарвардскую архитектуру, в соответствии с которой разделены не только адресные пространства памяти программ и памяти данных, но и шины доступа к ним. Каждая из областей памяти данных (оперативная память и EEPROM) также расположена в своем адресном пространстве.

Память программ (Flash ROM или Flash ПЗУ) используется для хранения последовательности команд, управляющих работой микроконтроллера, и имеет 16-ти битную структуру. Все AVR имеют Flash-память программ, различающихся по объему - от 1 до 256 КБайт. Ее главное

преимущество, это построение на принципе электрической перепрограммируемости, что допускает многократную перезапись информации. Программа записывается во Flash-память AVR как с помощью обычного программатора, так и с помощью SPI-интерфейса (Serial Peripheral Interface – последовательный периферийный интерфейс), в том числе непосредственно на собранной плате. Возможность внутрисхемного программирования (функция ISP – in-circuit serial programming) посредством интерфейса SPI имеют все микроконтроллеры AVR, кроме Tiny11 и Tiny28. Гарантированное число циклов перезаписи Flash-памяти у микроконтроллеров AVR второго поколения составляет не менее 10 тыс. циклов при типовом значении 100 тыс. циклов.

Память данных разделена на три части: регистровая память, оперативная память (ОЗУ – оперативное запоминающее устройство или RAM) и энергонезависимая память (ЭСПЗУ или EEPROM).

Регистровая память включает в себя 32 регистра общего назначения (РОН или GPR) и служебные регистры ввода/вывода (PBB). Они располагаются в адресном пространстве оперативной памяти, но не являются его частью. В области регистров ввода/вывода находятся служебные регистры (регистры управления микроконтроллером, регистры состояния и т. п.), а также регистры управления периферийными устройствами, входящими в состав микроконтроллера.

Для длительного хранения информации, которая может меняться в процессе работы микроконтроллера, используется EEPROM-память. Все AVR имеют блок энергонезависимой электрически перезаписываемой памяти данных (EEPROM) различающихся по объему - от 64 Байт до 4 КБайт. Данный тип памяти, используемый программой микроконтроллера в ходе ее выполнения, предназначен для хранения промежуточных данных, коэффициентов, констант, ключей и т.п. EEPROM может быть записана как через SPI интерфейс, так и с использованием программатора. Число циклов перезаписи составляет не менее 100 тысяч [2, 8, 9, 17, 18, 27].

Внутренняя оперативная статическая память (SRAM – Static RAM) имеет байтовый формат и служит для оперативного хранения данных. Ее размер, в зависимости от чипа, составляет от 64 Байт до 4 Кбайт. Количество циклов чтения и записи в оперативной памяти не ограничено, но при отключении питания вся информация теряется.

Периферия микроконтроллеров AVR включает: порты (от 3 до 48 линий ввода и вывода), поддержку внешних прерываний, таймеры-счетчики, сторожевой таймер, аналоговые компараторы, 10-разрядный 8-канальный аналого-цифровой преобразователь, различные интерфейсы связи, широтно-импульсные модуляторы. Порты ввода/вывода (I/O) AVR имеют число независимых каналов "вход/выход" от 3 до 53. Каждая линия порта может быть запрограммирована на вход или на выход. Особенность архитектуры построения портов ввода/вывода у AVR состоит в том, что для каждого физического вывода существует 3 бита управления, а не 2, как у распространенных 8-разрядных микроконтроллеров (Intel, Microchip, Motorola и т. д.). Благодаря чему пропадает необходимость хранить копию содержимого порта в памяти для безопасности и увеличивает быстродействие микроконтроллера при работе с периферийными устройствами, особенно при наличии внешних электрических помех.

Микроконтроллеры AVR работают при напряжениях питания от 1,8 до 6,0 Вольт. Ток потребления в активном режиме зависит от величины напряжения питания и частоты, на которой работает микроконтроллер, и составляет менее 1 мА для 500 кГц, 5 ... 6 мА для 5 МГц и 8 ... 9 мА для частоты 12 МГц. AVR могут быть переведены программным путем в один из трех режимов пониженного энергопотребления [2, 8, 9, 18, 25, 27]. Структура микроконтроллера на ядре AVR представлена на рисунке 2.

Интернет и его сервисов, таких как: электронные библиотеки, электронная коммерция, облачные технологии и других.

Затем развитие облачных технологий остановилось до 90-х годов, когда ряд факторов спровоцировал их дальнейшее развитие:

1) В 90-е годы ни технологии, ни компании не были готовы к увеличению пропускной способности сети Интернет, что не способствовало значительному скачку развития облачных технологий. Тем не менее, расширение пропускной способности сети Интернет предоставило ресурсы для дальнейшего совершенствования облачных вычислений.

2) Одним из ключевых событий в области облачных технологий стало возникновение в 1999 году компании Salesforce.com. Эта компания первой предоставила доступ к своему программному обеспечению через сайт. Тем самым, став первой компанией распространявшей свое приложение по принципу программное обеспечение как сервис (SaaS).

3) Дальнейшее развитие облачные технологии получили в 2002 году. Корпорация Amazon разработала свой облачный сервис, позволяющий производить вычисления, хранить и обрабатывать информацию

4) Следующим шагом являлся сервис Elastic Compute cloud (EC2) запущенный компанией Amazon. Данный сервис позволял пользователям запускать собственные приложения в облаке. Таким образом, сервисы Amazon являлись первыми общедоступными сервисами облачных вычислений.

5) Одним из важнейших событий в отрасли облачных технологий стала разработка корпорацией Google платформы Google Apps для web-приложений в бизнес-сегменте.

6) Большое значение в эволюции облачных технологий послужили технологии виртуализации, а именно программное обеспечение, служащее для создания виртуальной инфраструктуры.

7) Совершенствование аппаратного обеспечения, такое как увеличение емкости запоминающих устройств и появление многоядерных процессоров

сделало облачные технологии доступными широкому кругу пользователей, как частным лицам, так и малому бизнесу, что положительно повлияло на развитие отрасли [34].

1.3.1. Облачные сервисы и их типы.

Облачным вычислениям можно дать следующее определение. Облачные вычисления (англ. cloud computing) – модель обеспечения удобного сетевого доступа по требованию к некоторому общему фонду конфигурируемых вычислительных ресурсов. Например, сетям передачи данных, серверам, устройствам хранения данных, приложениям и сервисам – как вместе, так и по отдельности, которые могут быть оперативно предоставлены и освобождены с минимальными эксплуатационными затратами или обращениями к провайдеру

Достоинства облачных вычислений:

Доступность – возможность доступа из любого места обеспеченного доступом к сети Интернет, при помощи любого устройства имеющего браузер, что снижает затраты на приобретение дорогостоящей техники, программного обеспечения и обеспечивает мобильность пользователя.

Сравнительно низкая стоимость – уменьшаются расходы на содержание ИТ-инфраструктуры предприятия, в том числе сокращение штата обслуживающего персонала. К снижению расходов также ведет возможность фактического использования облачных ресурсов – пользователь оплачивает только необходимые ему вычислительные мощности. Уменьшаются затраты на приобретение дорогостоящего оборудования, в том числе за счет развития аппаратных ресурсов информационных технологий.

Гибкость – при использовании систем виртуализации, операции расширения и администрирования «облака» становятся более простыми. Вычислительные ресурсы неограниченны, при необходимости облачный сервис предоставляет требуемые для выполнения поставленной задачи

ресурсы (память, процессорное время, хранилище данных). При этом оплата производится только за фактически используемые мощности.

Надежность – как правило, облачные сервисы располагаются в специализированных центрах обработки данных, с высокой степенью надежности. Данные центры обеспечены защитой от компьютерных атак извне, функциями резервирования данных, резервным питанием, профессиональным обслуживающим персоналом и т. д.

Недостатки облачных вычислений:

Соединение с сетью – для пользования облачным сервисом обязательно постоянное подключение к сети Интернет.

Ограничения по программному обеспечению – существуют ограничения по размещению приложений в облачном сервисе и предоставления к ним доступа пользователю. Пользователи ограничены в выборе используемого программного обеспечения и настройки его под свои цели.

Конфиденциальность и надежность – существует возможность несанкционированного доступа к облачным данным. В настоящее время технологии, гарантирующей абсолютную защиту информации, не существует. Облачные сервисы могут быть подвержены как компьютерным атакам, так и вирусам, в случае использования сервисом стандартных операционных систем (Windows, Linux и др.).

Высокая стоимость оборудования – для создания собственного облака предприятию необходимо затратить большие материальные ресурсы, что недоступно небольшим компаниям [16, 34].

1.3.2. Модели обслуживания, предоставляемые облачными сервисами.

Все как услуга (Everything as a Service) модель, когда потребителю предоставляется готовое решение от программно-аппаратного комплекса, до управления бизнес-процессами.

Инфраструктура как услуга (Infrastructure as a service) пользователю предоставляется возможность использования облачной инфраструктуры для самостоятельного управления ресурсами обработки. Например, предоставляются возможности установки и использования любого программного обеспечения (операционные системы, прикладное программное обеспечение), контроля операционных систем, виртуальных хранилищ данных и сетевых сервисов.

Платформа как услуга (Platform as a service) предоставляется возможность использования настроенной облачной инфраструктуры для размещения и разработки программного обеспечения. В состав платформы могут входить среды разработки, тестирования и запуска прикладных программ.

Программное обеспечение как услуга (Software as a service) модель предоставляет возможность использования прикладного программного обеспечения облачного провайдера, доступного с разнообразных устройств пользователя, подключенных к сети интернет, посредством браузера, тонкого клиента или web-интерфейса программы. К таким услугам относят: сервисы электронной почты, онлайн органайзеры, онлайн системы документооборота.

Оборудование как услуга (Hardware as a Service) предоставляет возможность использовать оборудование провайдера как свое собственное, на котором строится клиентская инфраструктура с применением необходимого пользователю программного обеспечения.

Данные как услуга (Data as a Service) предоставляет потребителю дисковое пространство для хранения больших объемов данных [34].

1.3.3. Модели развертывания облачных сервисов.

Выделяют четыре модели развертывания облачных сервисов:

1. Публичные;
2. Частные;

3. Общественные;

3. Гибридные.

Публичное облако (public cloud) – инфраструктура, направленная на свободное использование облачного сервиса широким кругом пользователей. При этом потребители не имеют доступа к управлению и обслуживанию сервисом, эти операции выполняются поставщиком услуги. Например: Google Apps, Google Docs, Microsoft Office Online.

Частное облако (private cloud) – инфраструктура, используемая одной организацией и имеющая несколько потребителей (например: подразделения компании, клиенты компании).

Общественное облако (community cloud) – инфраструктура, предназначается для использования группой организаций из определенного сообщества, имеющих общие интересы и задачи (безопасность, соблюдение юрисдикции, политика, соответствие различным требованиям и т. д.).

Гибридное облако (hybrid cloud) – комбинация из двух или более разных облачных инфраструктур (общественных, частных, публичных). Которые остаются уникальными объектами, но связаны стандартизованными или частными технологиями передачи данных и приложений (например, временное использование ресурсов публичных облаков для выравнивания нагрузки между облаками) [34].

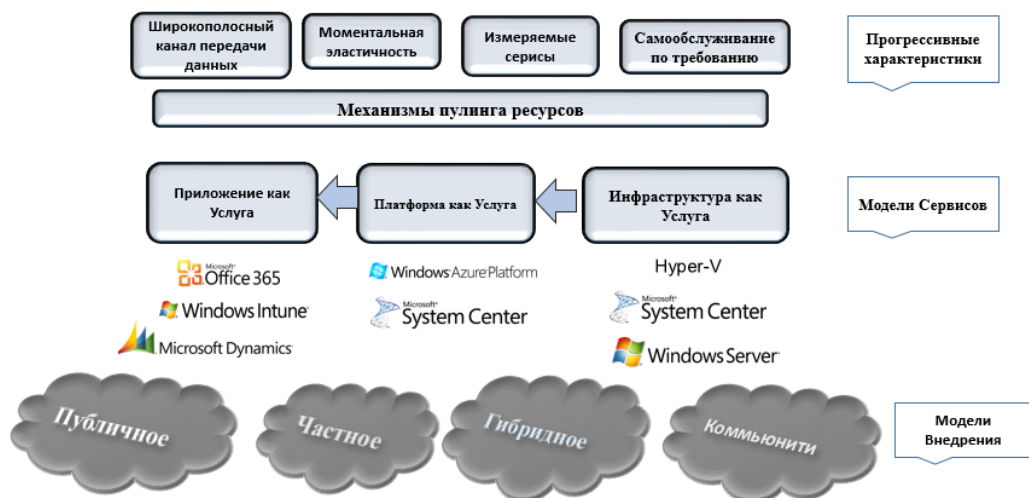


Рис. 3. Структура облачных вычислений.

Глава 2. Разработка системы контроля микроклимата в офисном помещении.

2.1. Архитектура аппаратных средств.

Выделяют два типа архитектуры системы управления помещением: централизованная и распределенная. Централизованная система состоит из центрального контроллера с подключенными к нему модулями. В распределенных системах управления устройства не зависят друг от друга. В этой архитектуре используется принцип построения системы на шине.

Достоинства централизованных систем:

1) Возможность создавать сложные системы управления. Центральный контроллер имеет достаточно производительности для работы с подключенными к нему датчиками.

2) Высокая скорость обработки информации. Сбор и обработка данных ведется непосредственно на контроллере.

3) Модули (датчики, исполнительные устройства) малогабаритны, недороги и имеют несложную техническую реализацию.

Недостатки централизованных систем:

1) Главным недостатком это сравнительно низкая надежность. При отказе центрального контроллера, функционирование системы прекращается.

2) Относительная дороговизна центрального модуля.

Достоинства распределенных систем:

1) Высокая надежность системы, так как при отказе одного или нескольких модулей система продолжает функционировать.

2) Простота масштабирования. К имеющейся шине возможно подключить дополнительные модули.

Недостатки распределенных систем:

1) Модули (датчики, исполнительные устройства) снабжены собственными контроллерами обработки информации, в результате чего достаточно сложны и недешевы.

2) Пониженная, по сравнению с централизованными системами, скорость обработки данных, так как данные обрабатываются в различных модулях [14, 16, 23, 24].

Архитектура системы управления:

Уровень 1: пользовательские системы управления: web-интерфейс, мобильные устройства, с помощью которых производится управление системой и отслеживание показаний с датчиков.

Уровень 2: центральный контроллер (ЦК), позволяет пользовательской системе вести взаимодействие с используемыми датчиками и исполнительными устройствами.

Уровень 3: датчики и периферийные устройства, снимают данные о состоянии окружающей среды (Рис. 4).



Рис. 4. Архитектура системы управления.

Алгоритм работы системы контроля:

Отслеживание состояния окружения происходит на основании данных полученных с датчиков. Через заданные периоды времени, либо при необходимости (например, срабатывание датчика движения), информация с датчиков передается на центральный контроллер, который её обрабатывает и осуществляет передачу данных в систему управления. Пользовательская система управления, это программно-аппаратный комплекс, позволяющий отслеживать микроклимат в помещении.

2.2. Программно-аппаратные средства для системы контроля микроклимата.

Исходя из целей, заданных для преддипломной практики, были выбраны следующие элементы:

- 1) Микроконтроллер ATmega328 на платформе Arduino Uno в качестве центрального контроллера.
- 2) Датчик измерения влажности и температуры воздуха DHT11.
- 3) Пирозлектрический инфракрасный (PIR) датчик на чипе BISS0001.

2.2.1 Микроконтроллер.

Arduino Uno выполнен на микроконтроллере (МК): ATmega328. Платформа имеет 14 цифровых входов/выводов, 6 аналоговых входов, кварцевый генератор 16 МГц, USB порт, разъем питания, разъем ICSP. Для работы требуется подключить платформу к компьютеру с помощью USB кабеля, либо подать питание с адаптера AC/DC, или батареи. Принципиальная схема устройства представлена в приложении 1.



Рис. 5. Внешний вид платформы Arduino Uno.

Работа платформы возможна при внешнем питании от 6 В до 20 В. При напряжении питания ниже 7 В, вывод 5V может выдавать менее 5 В, при этом возможна нестабильная работа системы. При использовании напряжения выше 12 В возможен перегрев регулятора напряжения, выход платы из строя. Рекомендуемый диапазон от 7 В до 12 В.

Выводы питания:

VIN. Вход служит для подачи питания от внешнего источника (в отсутствие 5 В от разъема USB или другого источника питания). Напряжение питания производится через данный вывод.

5V. Регулируемый источник напряжения, используется для питания МК и компонентов на плате. Питание может поступать от вывода VIN, от разъема USB, либо другого регулируемого источника напряжения 5 В.

3V3. Напряжение на выводе 3.3 В генерируется микросхемой FTDI. Максимальное потребление тока 50 мА.

GND. Выводы заземления.

Память:

Микроконтроллер ATmega328 имеет 32 кБ флэш-памяти для хранения кода программы (при этом 0,5 кБ используются для хранения загрузчика). ATmega328 имеет 2 кБ ОЗУ и 1 Кб EEPROM.

Входы и Выходы:

Каждый из 14 цифровых контактов Arduino Uno работает при напряжении 5 В и может настраиваться как вход или выход с помощью функций `pinMode()`, `digitalWrite()`, и `digitalRead()`. Каждый вывод имеет нагрузочный резистор (стандартно отключен) 20-50 кОм и может пропускать ток до 40 мА. Некоторые выводы имеют особые функции:

Последовательная шина: 0 (RX) и 1 (TX). Используется для получения (RX) и передачи (TX) данных TTL. Данные выводы подключены к соответствующим выводам микросхемы последовательной шины FTDI USB-to-TTL.

Внешнее прерывание: 2 и 3 выводы могут быть настроены для вызова прерывания либо на младшем значении, либо на переднем или заднем фронте, или при изменении значения.

ШИМ: 3, 5, 6, 9, 10, и 11, любой из этих выводов обеспечивает ШИМ с разрешением 8 бит при помощи функции `analogWrite()`.

LED: 13. Встроенный светодиод, подключенный к цифровому выводу 13. Если значение на выводе имеет высокий потенциал, то светодиод горит.

Связь:

Платформа Arduino Uno имеет несколько устройств для осуществления связи с персональным компьютером (ПК), другими устройствами Arduino либо микроконтроллерами. ATmega328 поддерживает последовательный интерфейс UART TTL (5 В), осуществляемый выводами 0 (RX) и 1 (TX). Установленная на плате микросхема FTDI FT232RL направляет данный интерфейс через USB, а драйверы FTDI (включены в программу Arduino) предоставляют виртуальный COM-порт программе на компьютере. Мониторинг COM-порта (Serial Monitor) программы Arduino предназначен для получения и отправки текстовых данных при подключении к платформе.

Стандартный способ считывания данных с Arduino состоит в подключении этого устройства через USB интерфейс к персональному компьютеру, при этом создается виртуальный COM-порт для обмена информацией. При использовании в качестве источника питания батареи или

адаптера AC/DC, а так же при наличии альтернативного канала связи (например: LAN, Wi-Fi, Bluetooth), платформа может работать без непосредственного подключения к ПК. Так же поддерживается подключение плат расширения (Shield), которые предоставляют дополнительный перечень возможностей [1, 3, 4, 8, 10, 27].

Таблица 1

Характеристики платформы Arduino Uno

Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7-12 В
Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА
Флеш-память	32 Кб, при этом 0.5 Кб используются для загрузчика
ОЗУ	2 Кб
EEPROM	1 Кб
Тактовая частота	16 МГц

2.2.2. Используемые датчики.

Arduino поддерживает большой перечень датчиков. В данной работе использованы распространенные датчики DHT11 (применяется для получения показателей температуры и относительной влажности воздуха) и пирозлектрический инфракрасный (PIR) датчик на чипе BISS0001.

PIR датчики служат для улавливания движения. Получили широкое распространение в системах сигнализации. Эти датчики малы по размерам, дешевы, энергоэффективны, просты в использовании и долговечны.



Рис. 6. Пассивный инфракрасный датчик (PIR).

Общая техническая информация:

- 1) Выходной сигнал: цифровой импульс HIGH (3 В) при наличии движения и цифровой сигнал LOW, при его отсутствии.
- 2) Дальность работы датчика: До 6 метров (110° x 70° область обнаружения)
- 3) Рабочее напряжение: 5 - 9В;
- 4) Питание: 3В - 9В, оптимальное напряжение использования - 5 вольт;

Принцип работы PIR датчиков движения: датчик состоит из двух основных частей. Они включают в себя материал, чувствительный к инфракрасному излучению. При нахождении датчика в состоянии покоя, обе части фиксируют одинаковое количество излучения. Например, излучение помещения или окружающей среды на улице. При пересечении зоны чувствительности первого сенсора теплокровным объектом (человек или животное), на модуле датчика генерируются два различных значения. Когда объект покидает зону чувствительности сенсора, значения выравниваются.

Изменения в показаниях обоих датчиков фиксируются, и происходит генерация импульса на выходе (HIGH или LOW) [28, 30, 35].

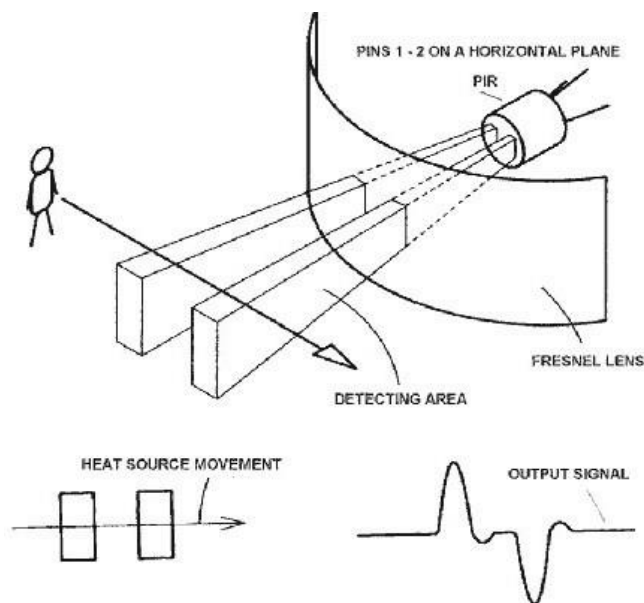


Рис. 7. Схема работы ИК датчика.

Модуль имеет 3 вывода (стандарта 2.54мм):

- 1) GND: "-" питание;
- 2) OUT: Вывод выходного сигнала;
- 3) VCC: "+" питание.

Подключение датчика :

GND на любой из GND выводов Arduino;

OUT на любой из цифровых входов/выводов Arduino;

VCC на + 5 вольт на Arduino.

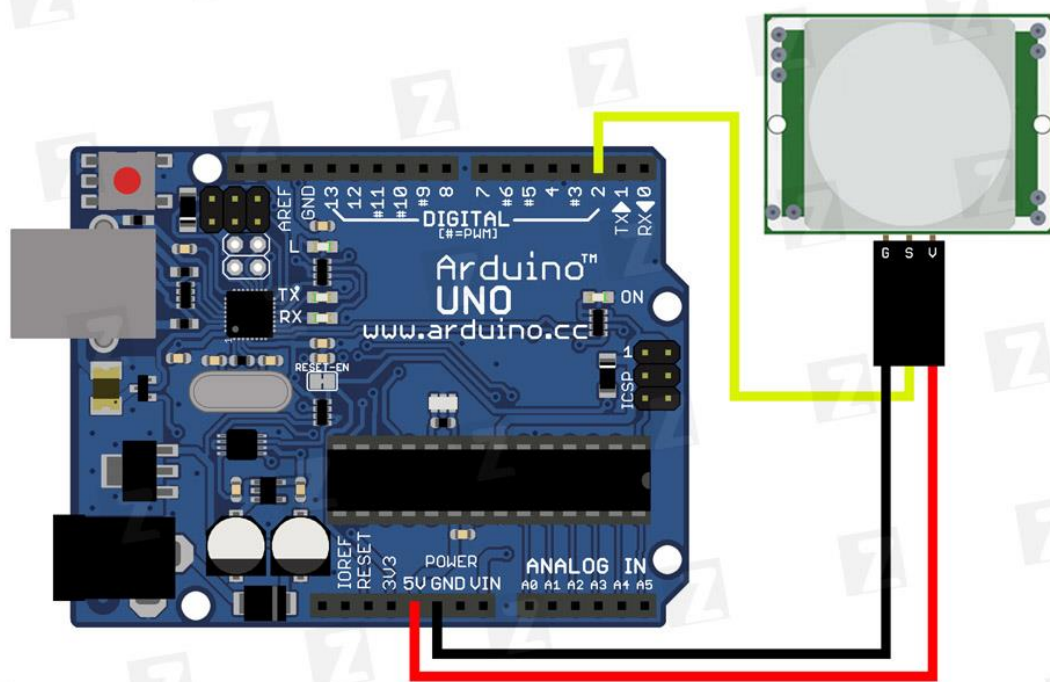


Рис. 8. Подключение ИК датчика.

Датчик измерения температуры и относительной влажности DHT11.

Основные технические характеристики:

- 1) Напряжение питания: 3 - 5В;
- 2) Определяемая влажность: 20 - 80% \pm 5%;
- 3) Определяемая температура: 0 - 50° \pm 2%;
- 4) Частота опроса: \leq 1Гц;
- 5) Размеры: 30 x 14 x 6мм.

Данный датчик не рассчитан на работу в экстремальных условиях, но для использования в помещении и для решения поставленной задачи его характеристик достаточно. Внутри датчика находится емкостной датчик влажности, термистор, и аналогово-цифровой преобразователь значений температуры и влажности.

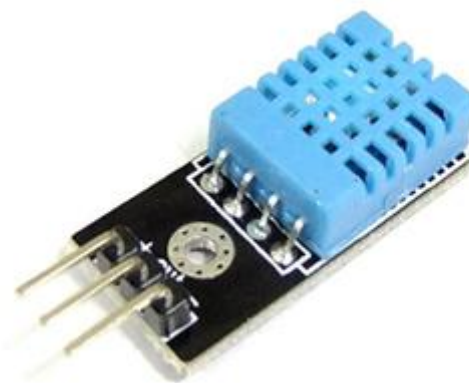


Рис. 9. Датчик DHT11.

Данные с датчика о температуре и влажности передаются в цифровом виде, что позволяет передавать информацию по соединительному кабелю длиной до 20 м. Так как измерения показателей окружающей среды выполняются по команде микроконтроллера, датчик является энергоэффективным [28, 30, 35].

Алгоритм работы датчика DHT11: при необходимости получения показателей с датчика, микроконтроллер посылает на сигнальный контакт значения ноль, затем единицу, чем подтверждает готовность принимать данные. Датчик производит замер и отправляет на вывод пятибайтную последовательность нулей и единиц. По два байта в передаче занимают показатели влажности и температуры воздуха. Оставшийся байт несет в себе контрольную сумму.

Подключение к Arduino:

Модуль имеет 3 вывода (стандарта 2.54мм):

G - Подключается к выводу GND;

V - Подключается к выводу +5V;

S - Подключается к любому цифровому выводу.

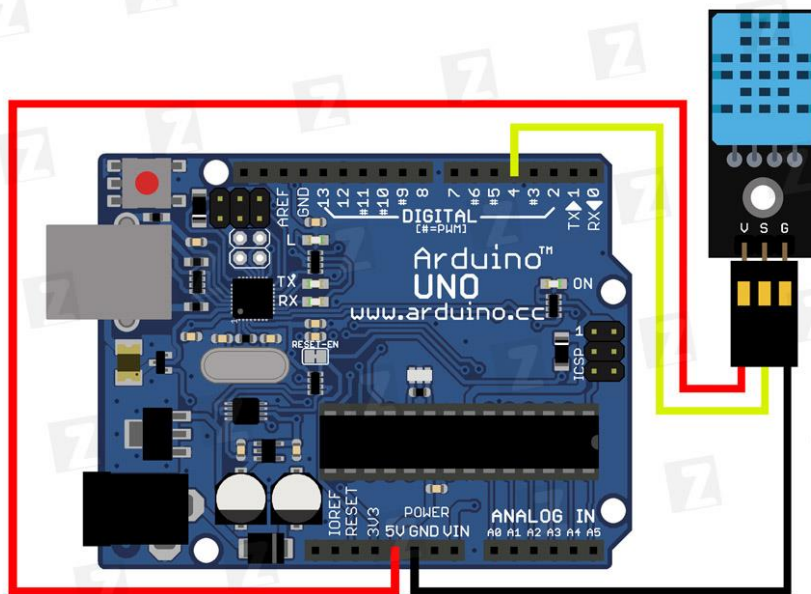


Рис. 10. Подключение датчика DHT11.

2.3. Программные средства, для системы контроля микроклимата в офисном помещении.

2.3.1. Среда программирования Arduino IDE.

В качестве программной среды разработки для платформы Arduino используется среда Arduino IDE. она отвечает всем необходимым требованиям: бесплатность; наличие русифицированного интерфейса; кроссплатформенность; простота установки и настройки; наличие различных библиотек расширяющих функционал.

Arduino IDE - среда со своим языком программирования разработанная компанией Arduino Software для программирования микроконтроллеров на платформе Arduino собственного производства. Язык программирования Ардуино основан на C/C ++, и предоставляет возможность использования всех его функций. Написанный код преобразуются (с некоторыми изменениям) в программу на языке C/C++, и далее компилируются компилятором AVR-GCC. Среда разработки Arduino

состоит из встроенного текстового редактора программного кода, области сообщений, окна вывода текста (консоли), панели инструментов с кнопками часто используемых команд и нескольких меню (Рис. 11). Программа, созданная в среде Arduino, называется эскиз (скетч). Эскиз пишется во встроенном текстовом редакторе, имеющем возможности вырезки, вставки, поиска и замены текста. При сохранении или экспорте программы в область сообщений выводятся пояснения, и отображаются возникающие ошибки. Окно вывода текста (консоль) выводит сообщения, полученные от микроконтроллера, включающие полные данные по ошибкам и другую информацию [2, 11, 14, 18, 22, 25].

Языки интерфейса Arduino IDE английский, русский. Программа поддерживает операционные системы семейства Windows, Linux и Mac OS X. Распространение программы: Freeware (бесплатная).

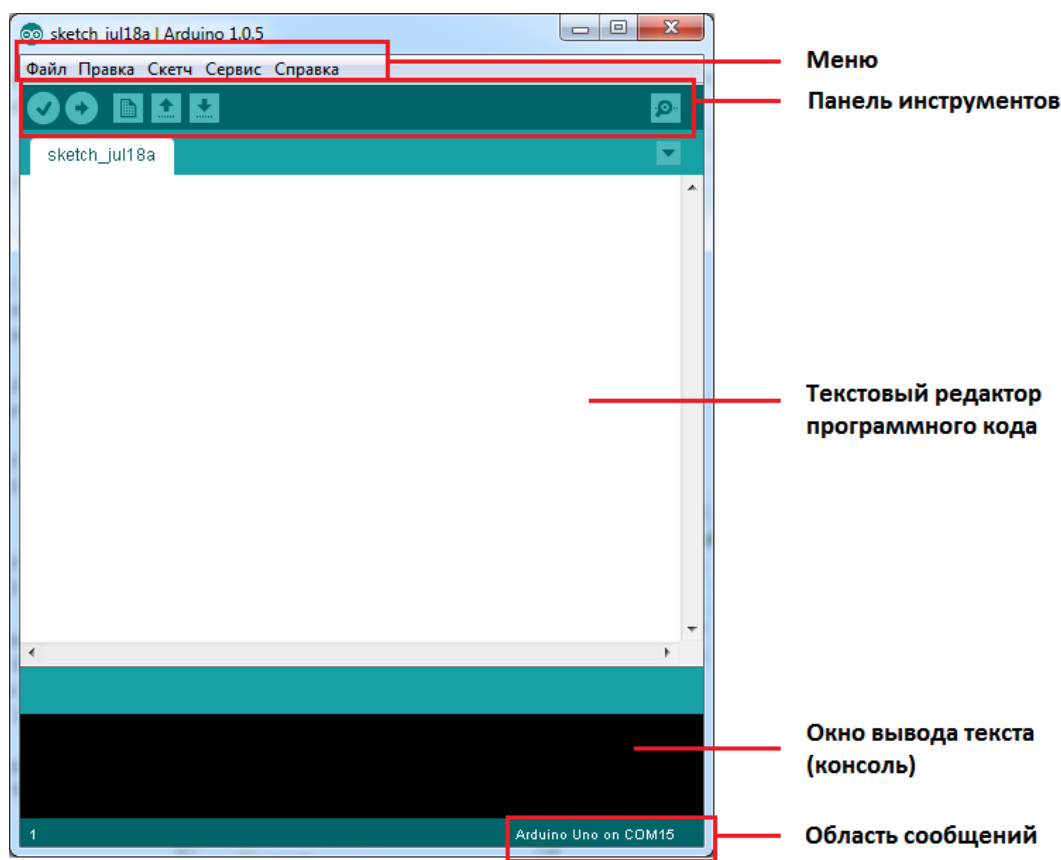


Рис. 11. Интерфейс Arduino IDE.

Меню «Файл» содержит стандартные пункты, такие как: создать, открыть, сохранить и другие, применяемые и в другом программном обеспечении. Пункт меню «Папка со скетчами» позволяет получить быстрый доступ ко всем созданным скетчам. Пункт меню «Примеры» содержит краткие скетчи, способные продемонстрировать работу с синтаксисом языка программирования и подключаемыми устройствами.

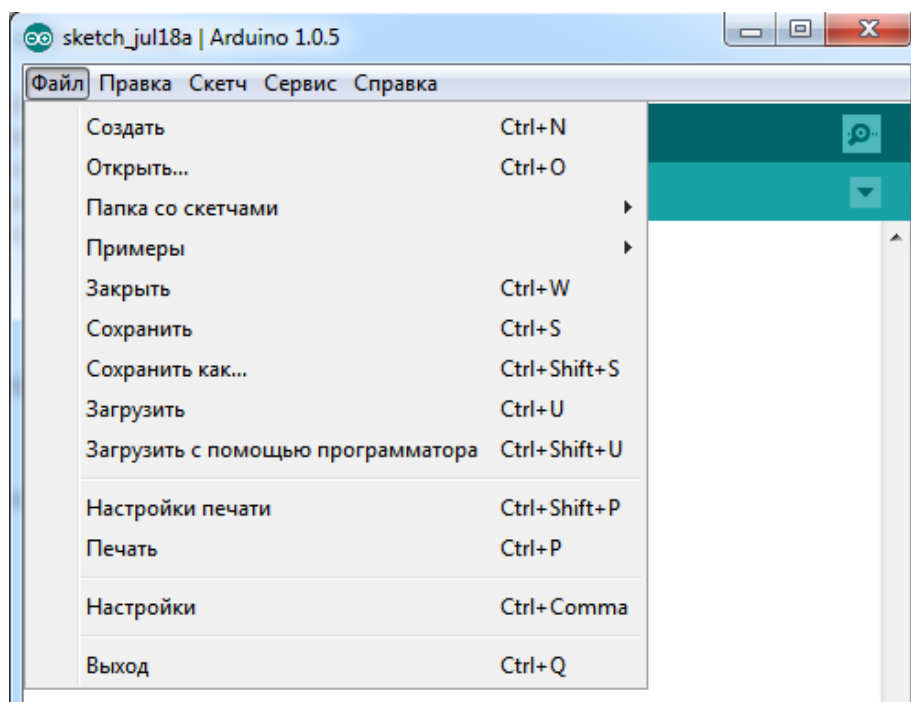


Рис. 12. Меню «Файл».

В меню «Правка» перечислены команды, используемые для редактирования кода написанного скетча, они удобны наличием комбинаций клавиш быстрого доступа к командам (Рис. 13).

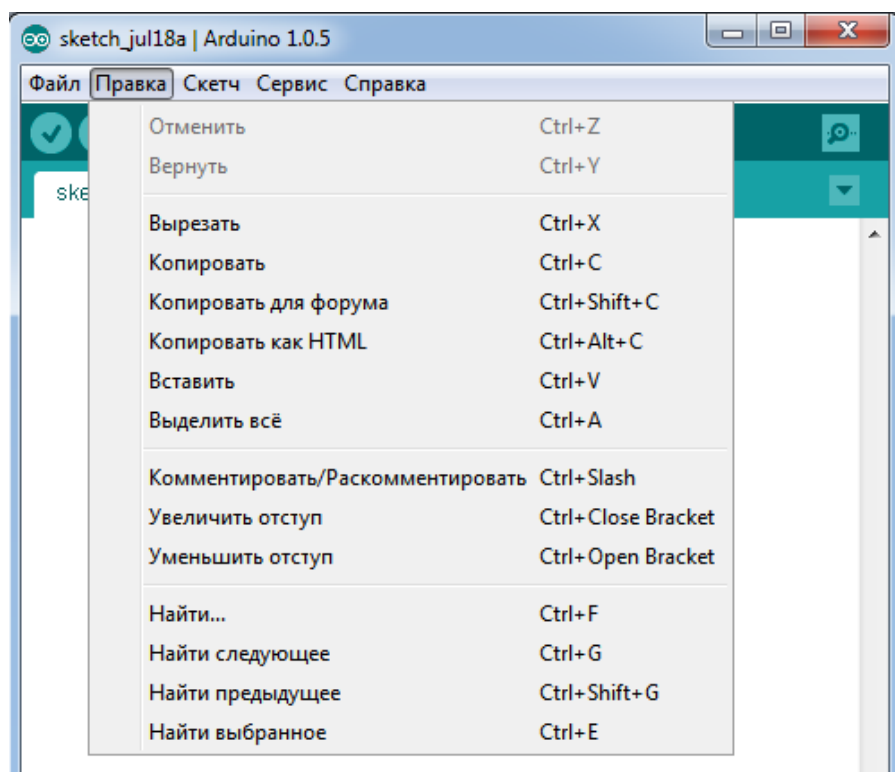


Рис. 13. Меню «Правка».

Меню «Скетч». В этом меню дублируется команда с панели управления «Проверить / Компилировать». При выборе которой происходит проверка кода на ошибки, и при успешном ее прохождении, компиляция программы.

Пункт «Показать папку скетчей» позволяет открыть каталог с текущим открытым кодом.

«Добавить файл...» открывает файл с кодом в отдельно создаваемой вкладке.

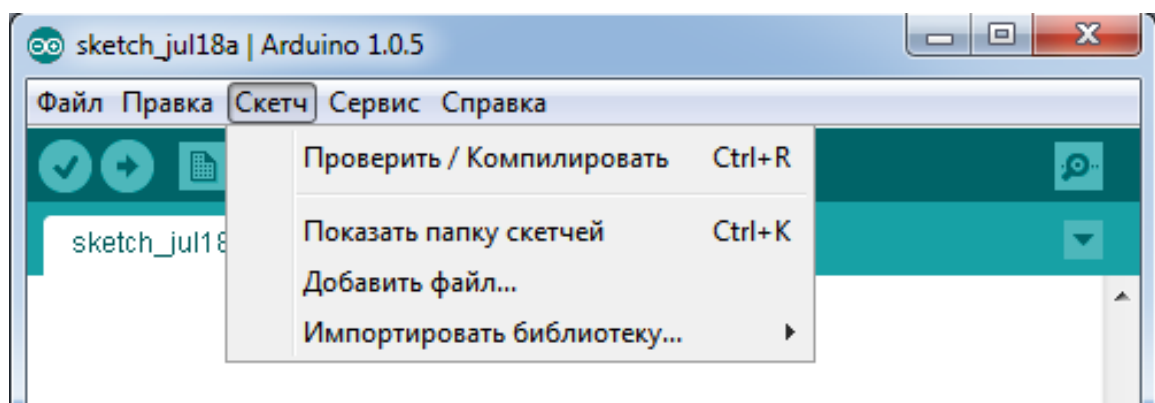


Рис. 14. Меню «Скетч».

«Импортировать библиотеку...» позволяет осуществить импорт и подключение внешней библиотеки в код программы. Внешние библиотеки необходимы при работе с подключаемыми к Arduino устройствами и датчиками, так как содержат инструкции для корректной работы с используемой периферией.

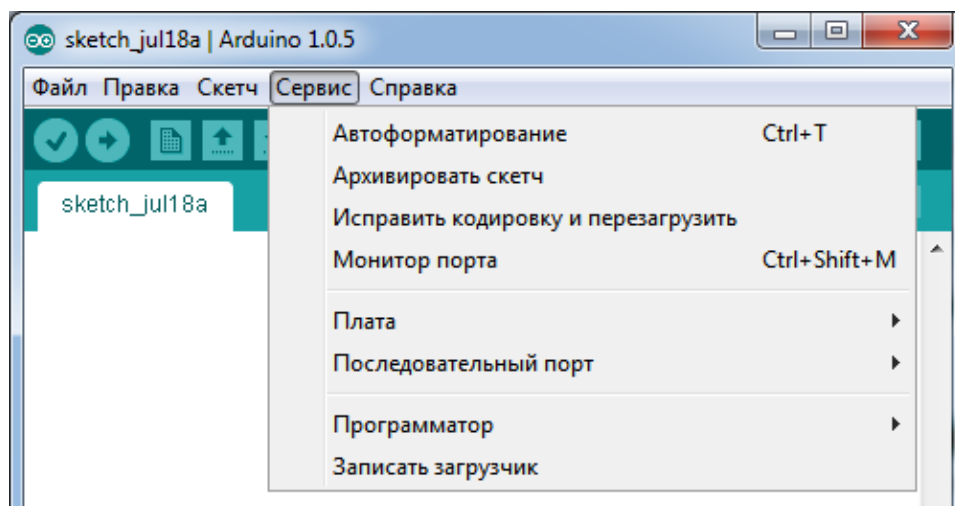


Рис. 15. Меню «Сервис».

В меню «Сервис» указывается модель платы Arduino подключенной к ПК и COM порт, используемый для передачи данных на контроллер.

Функция «Автоформатирование» служит для автоматического изменения форматирования кода и делает текст программы более удобным для чтения.

Среда Arduino IDE имеет встроенную возможность записать загрузчик на контроллеры Atmega: пункт меню «Программатор» определяет используемое устройство, «Записать загрузчик» производит запись загрузчика для указанной модели платы на микроконтроллер.

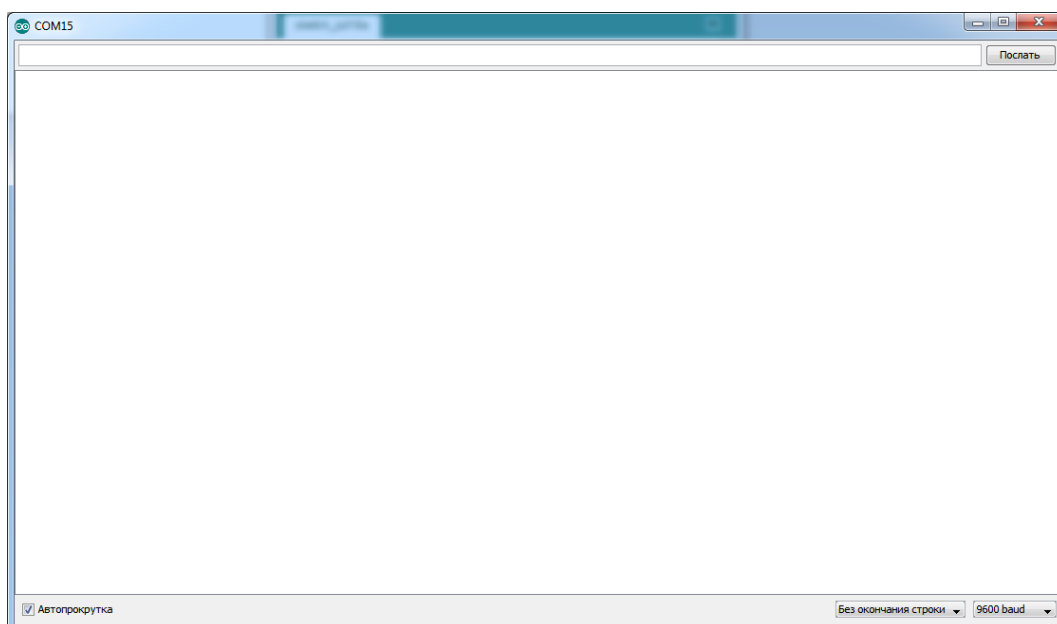


Рис. 16. Монитор порта.

Пункт «Монитор порта» открывает интерфейс обмена сообщениями с микроконтроллером посредством СОМ-порта (Рис. 16).

Меню «Справка», аналогично подобным меню в другом программном обеспечении, содержит справочную информацию по использованию программы.

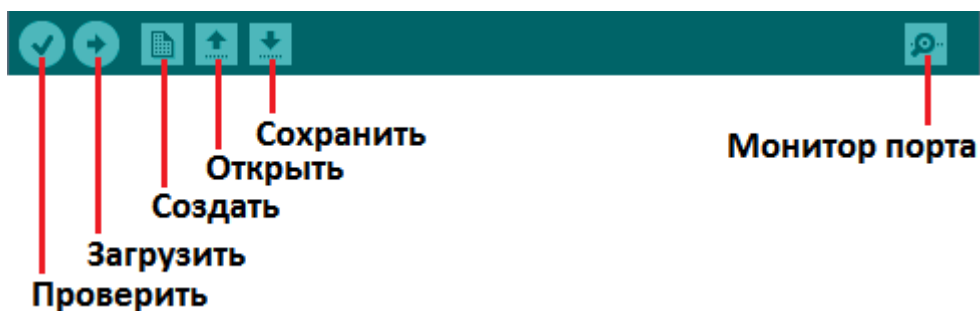


Рис. 17. Панель управления.

Кнопки на панели управления дублируют наиболее используемые команды из меню программы.

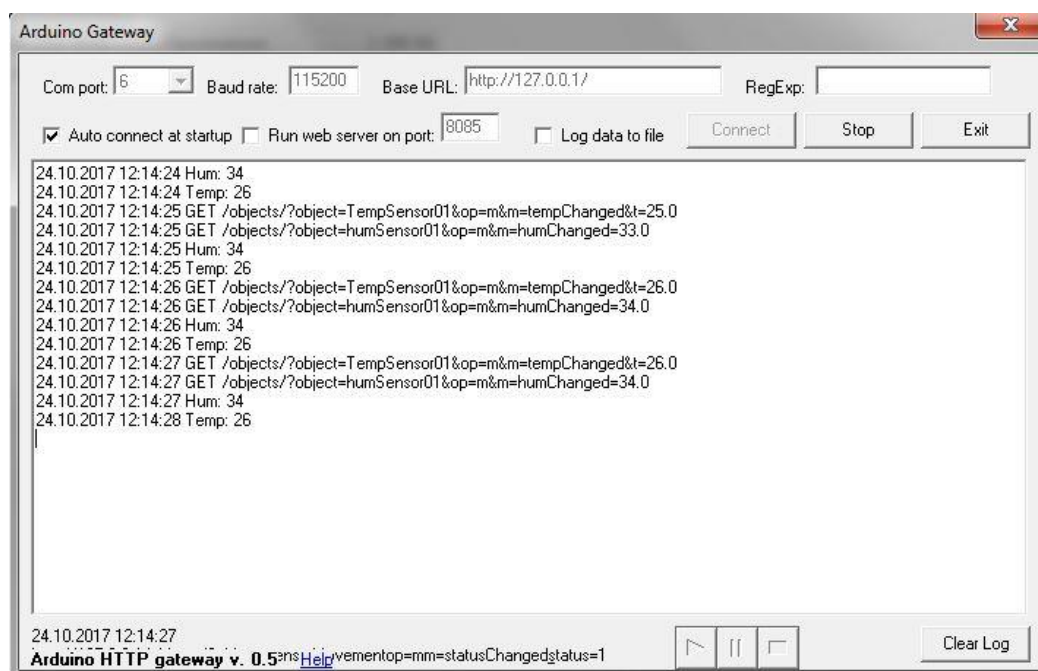


Рис. 18. Интерфейс Arduino Gateway.

Для передачи данных от микроконтроллера в систему управления применяем программу Arduino Gateway. Она позволяет считывать данные передаваемые микроконтроллером на COM порт и отправлять их на хост системы управления как HTTP запросы.

Интерфейс и настройки программы элементарны. Для работы необходимо указать номер COM-порта, скорость обмена данными и http-адрес, на который будут отправляться команды (Рис. 18).

2.3.2. Система домашней автоматизации MajorDoMo.

В качестве платформы для контроля помещения используется Система домашней автоматизации MajorDoMo.

Система домашней автоматизации MajorDoMo (Major Domestic Module или Главный Домашний Модуль) это бесплатная и открытая программная платформа для комплексного управления домашней автоматикой. Данную систему? возможно установить практически на любой персональный компьютер использующий платформы Windows или Linux.

Основные возможности MajorDoMo:

- Простая и быстрая установка;
- Кроссплатформенность (Windows/Linux);
- Бесплатность для личного и коммерческого использования;
- Поддерживает разнообразное оборудование;
- Наличие русифицированного интерфейса;
- Доступ с любого устройства через Web-интерфейс;
- Web-интерфейс с обновлением в реальном времени;
- Интеграция со сторонними веб-сайтами и сервисами;
- Модель безопасности с разграничением доступа между пользователями;
- CloudSync - облачная синхронизация и простой доступ из любого места;
- Удобная система обновлений [33].

Установка системы домашней автоматизации MajorDoMo.

Приложение является бесплатным и доступен для скачивания без ограничений с официального сайта <https://majordomo.smartliving.ru>. Установка стандартная для Windows приложений. При запуске программа установки предлагает выбрать язык, используемый при установке.

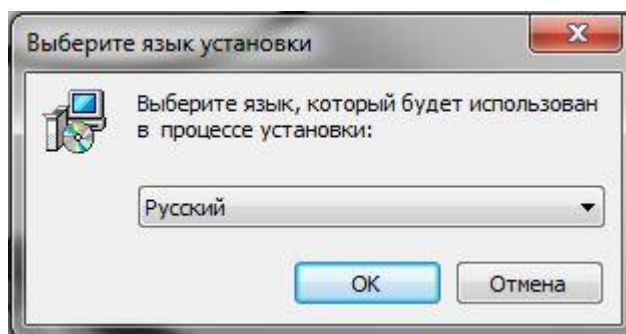


Рис. 19. Выбор языка установки.

Далее необходимо указать каталог для установки приложения на диск.

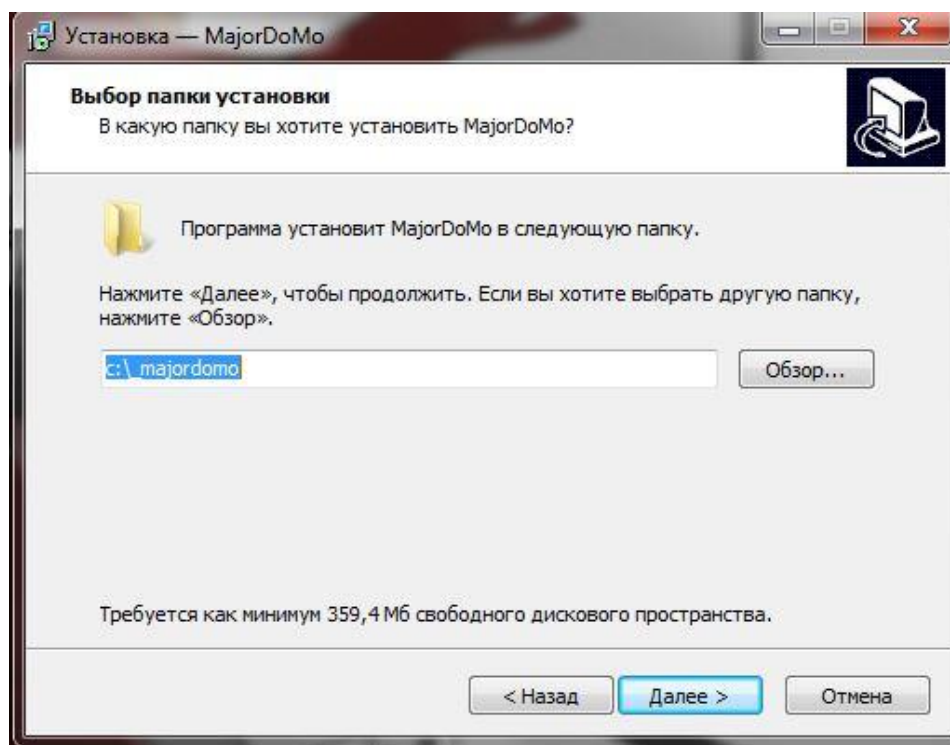


Рис. 20. Выбор каталога для установки.

Затем следует выбор устанавливаемых компонентов. Для стабильности работы системы рекомендуется устанавливать только ядро программы, обновления компонентов можно выполнить после установки ПО.

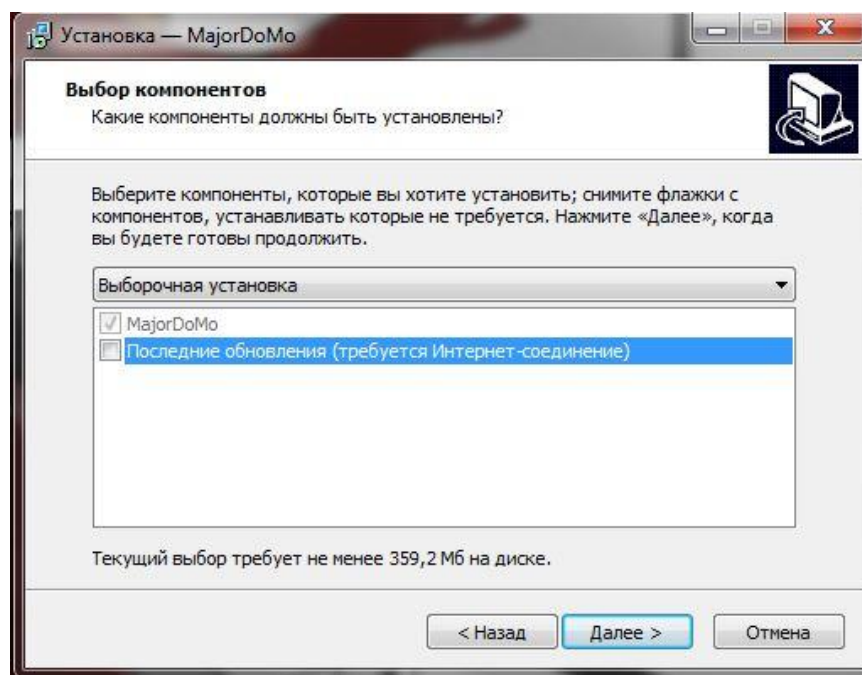


Рис. 21. Выбор компонентов ПО.

После окончания установки есть возможность сразу запустить приложение.

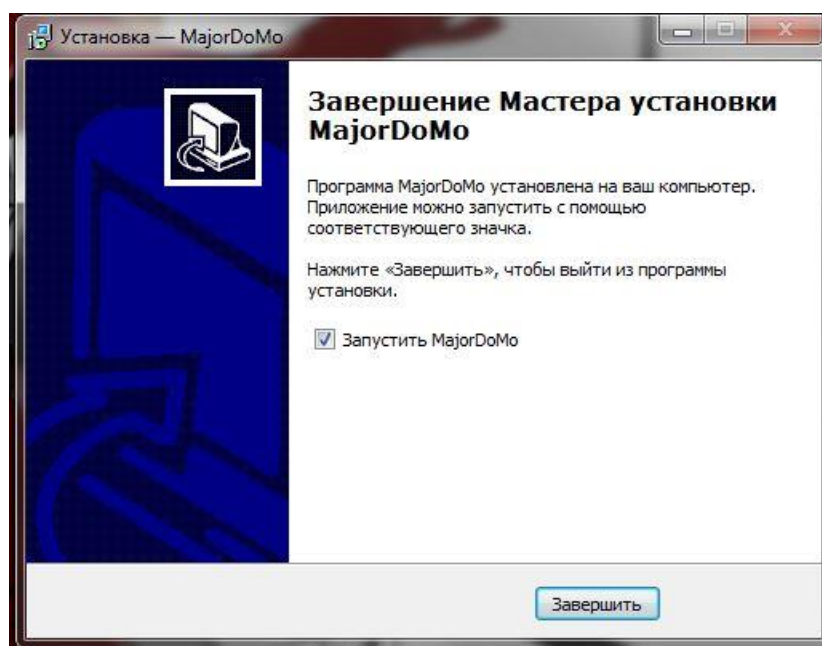


Рис. 22. Завершение установки.

При первом запуске системы запускается браузер с открытым Web-интерфейсом системы управления, в котором необходимо выбрать язык локализации приложения и тему оформления. При необходимости, это можно выполнить позднее в настройках программы.

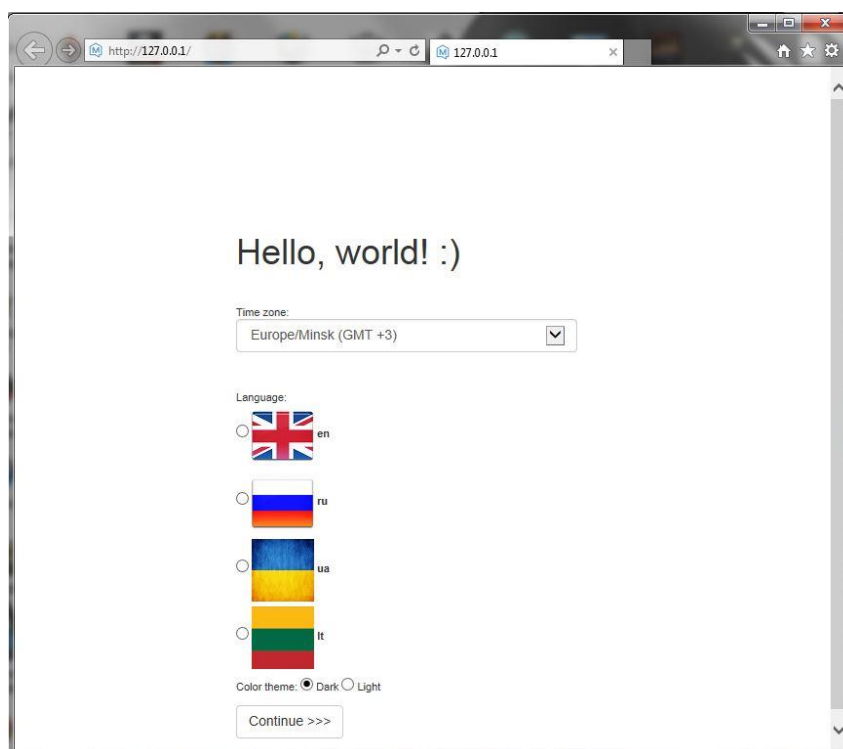


Рис. 23. Первый запуск системы управления.

Интерфейс системы управления доступен из любого браузера по адресу <http://127.0.0.1/> (Рис. 22). На главной странице отображаются полученные с внешних датчиков и устройств данные, в данном случае это: влажность и температура воздуха, а так же отслеживается движение в помещении. При необходимости возможно переключение на «сцену» виртуальную модель помещения с отображающейся на ней интерактивной информацией. При изменении показателей данные в системе управления обновляются мгновенно, что позволяет получить полную и актуальную информацию о состоянии микроклимата помещения.

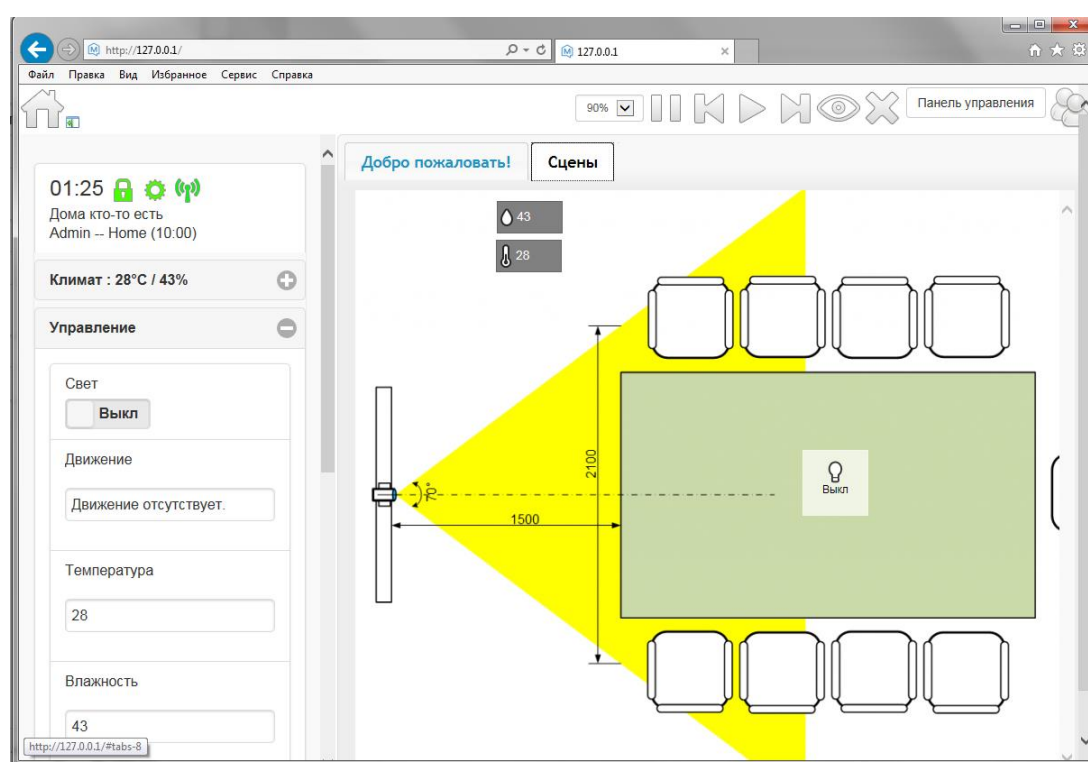


Рис. 24. Главное окно системы MajorDoMo.

Настройка и программирование системы управления осуществляется через панель управления доступной на главном экране.

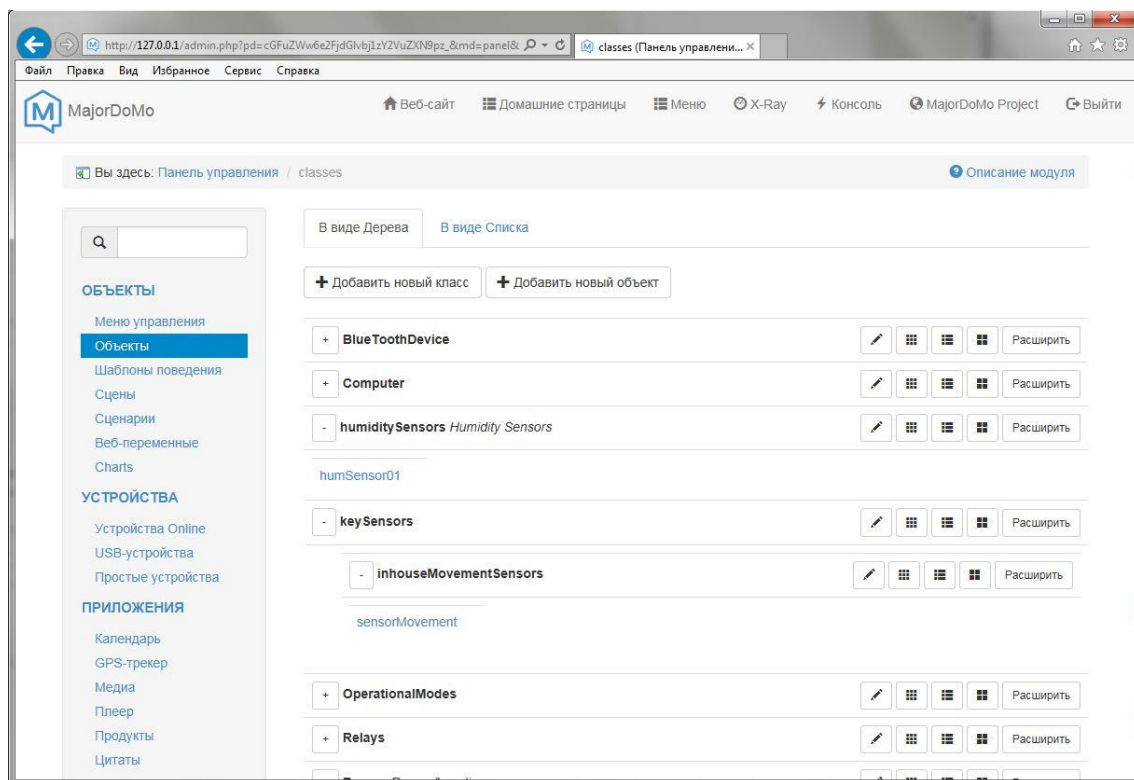


Рис. 25. Панель управления.

Панель управления обладает обширным функционалом, позволяющим добавлять неограниченное количество элементов, таких как датчики, исполнительные устройства, визуальные элементы и другие.

Система управления MajorDoMo позволяет взаимодействовать с большим количеством объектов, например, в данном случае настроено взаимодействие с датчиком влажности и температуры и датчиком движения. При получении http сообщения с данными от датчиков, отправленными микроконтроллером, система управления изменяет значения показателей и выводит их в удобном для пользователей виде: текстовым сообщением или пиктограммой с соответствующим значением.

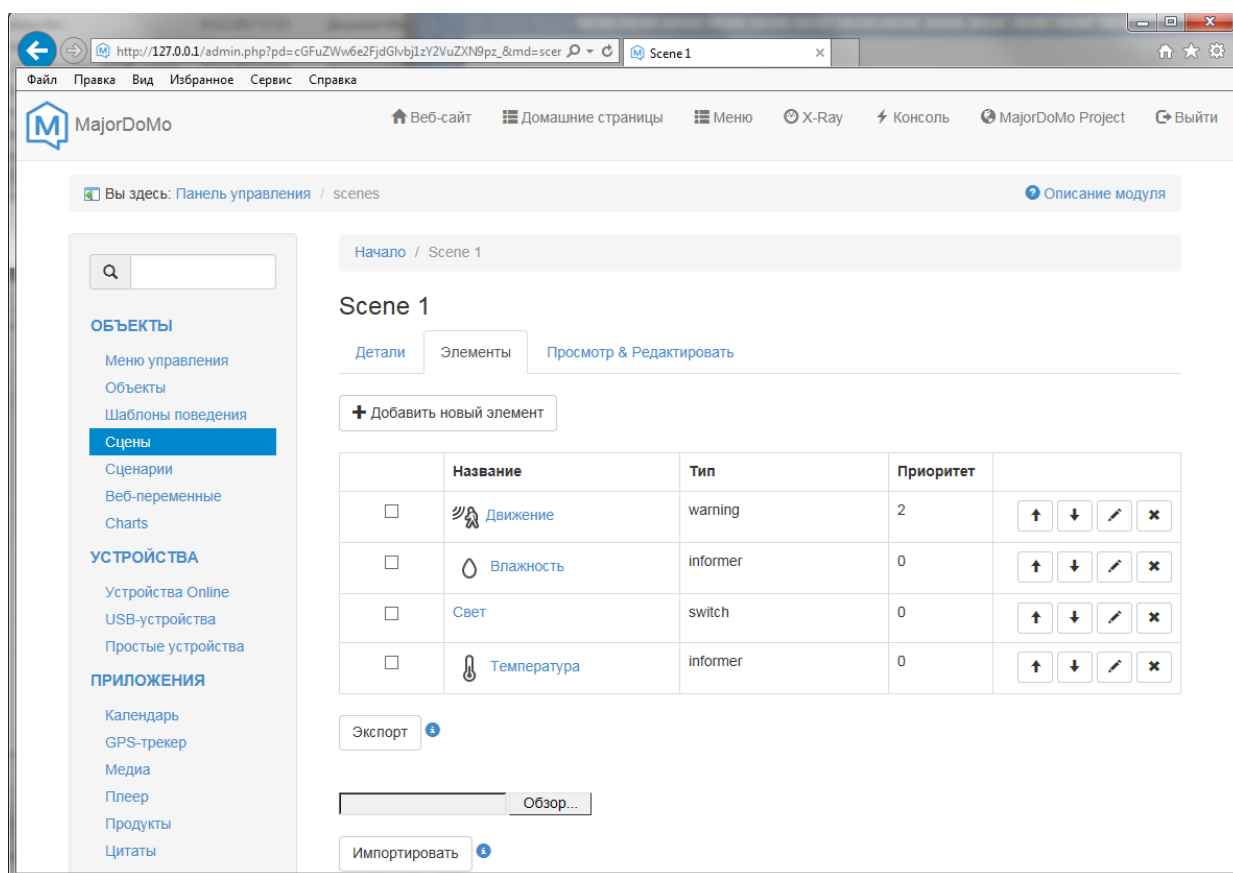


Рис. 26. Настройка элементов.

2.3.3. Сервис облачной синхронизации и управления CloudSync.

Одной из особенностей MajorDoMo является возможность доступа к системе из любого места при помощи облачного сервиса CloudSync. Сервис предоставляет услугу вида Information-as-a-Service ("информация как сервис") и дает возможность удаленно использовать любые виды информации, которая может меняться ежеминутно или даже каждую секунду и позволяет производить мониторинг данных без привязки к оборудованию или местоположению. Одной из главных особенностей данного сервиса является отсутствие необходимости настройки прямого подключения к устройству через Интернет, не требует дополнительной настройки маршрутизации и работает через большинство распространенных Firewall-ов.

Для использования CloudSync, необходимо зарегистрироваться на официальном сайте <https://connect.smartliving.ru/> и активировать CloudSync в MajorDoMo (раздел Connect в панели управления).

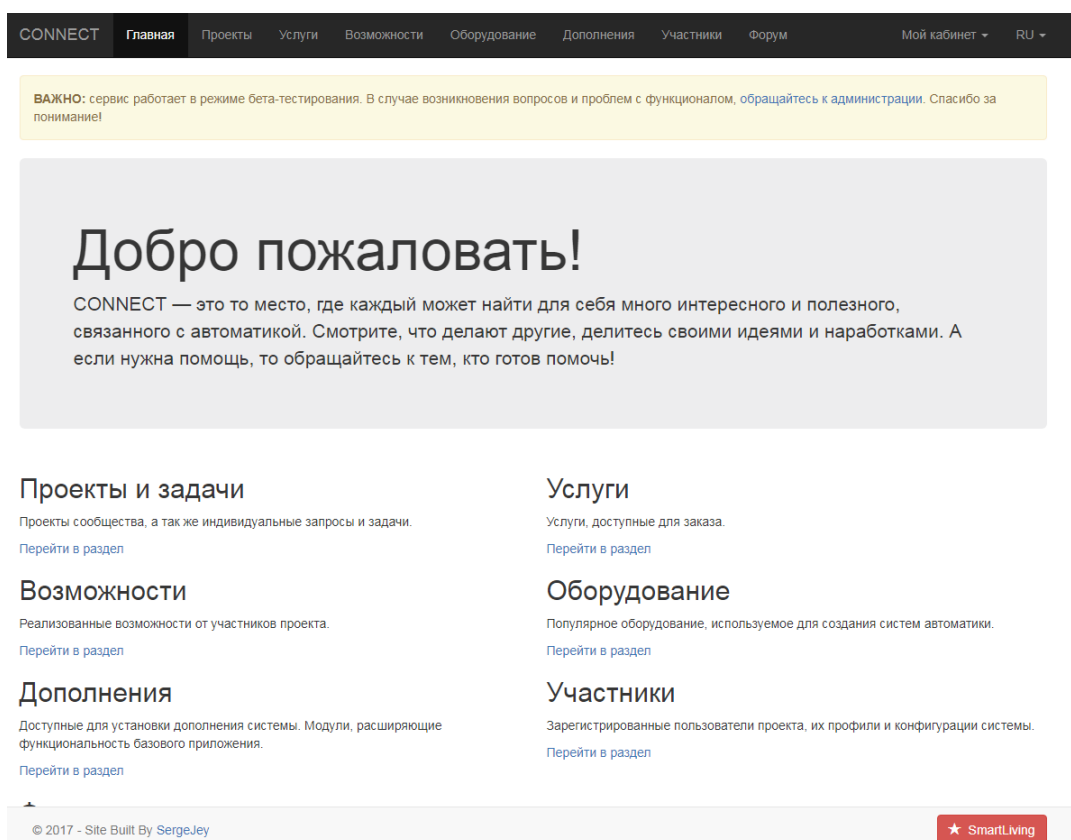


Рис. 27. Официальный сайт Connect.

Для системы MajorDoMo сервис предоставляет такие функции как синхронизация элементов меню и доступ к нему через интернет по ссылке <http://connect.smartliving.ru/menu.html>.

Интерфейс облачной системы управления интуитивно понятен и прост в работе. При этом он позволяет получать данные показателей датчиков и периферийных устройств. Также, при необходимости возможно удаленное управление исполнительными устройствами, при их наличии.

Благодаря данному сервису, мы получаем возможность контролировать микроклимат помещения из любого места, при наличии доступа в Интернет.

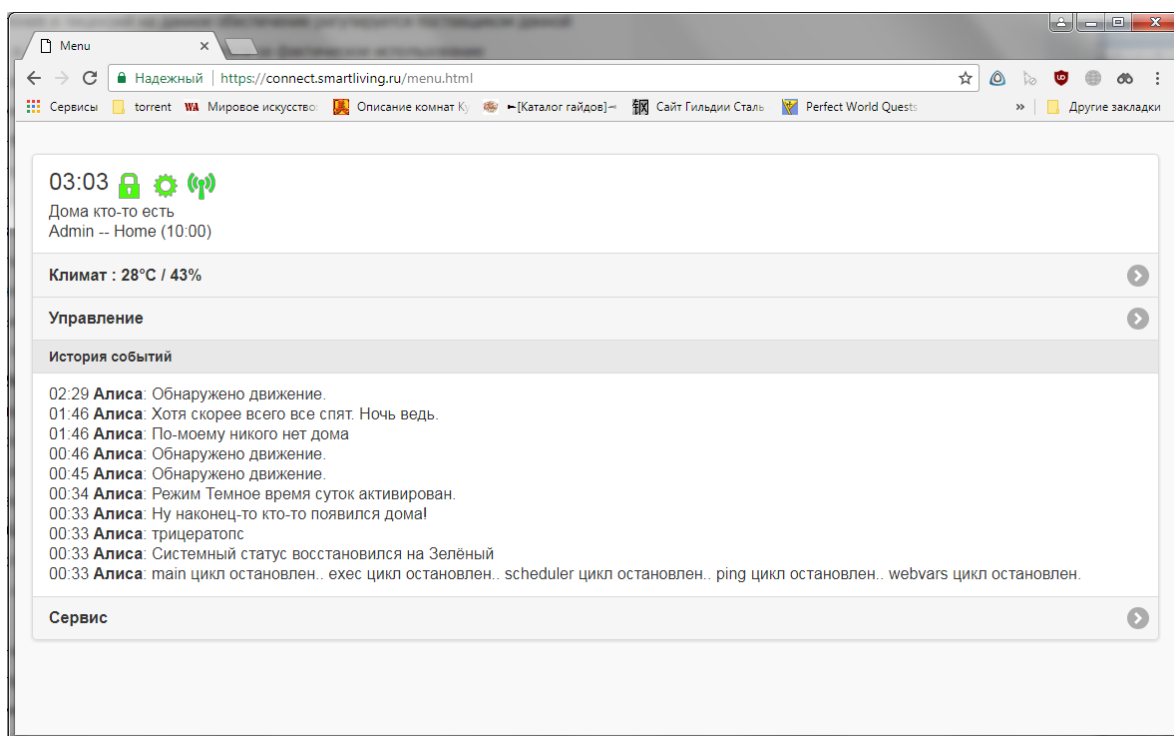


Рис. 28. Интерфейс облачной системы управления.

2.3.4. Программирование микроконтроллера.

Описание алгоритма программы:

В программе используется внешняя библиотека `dht11.h` для взаимодействия с датчиком влажности и температуры DHT11.

Операции опроса датчиков выполняются с использованием цикла «IF».

При подаче питания происходит инициализация и считывание информации с датчика движения. В случае обнаружения движения, на выходе PIR датчика формируется логический сигнал HIGH и микроконтроллер производит вывод данных о наличии движения на COM-порт. Информация с COM-порта преобразуется в http запрос и отправляется по адресу, на котором настроена система управления MajorDoMo.

Датчик температуры и влажности так же инициализируется и с периодичностью в пять секунд происходит запрос данных о текущих значениях температуры и влажности окружающей среды. Полученные данные так же, как и в случае с PIR датчиком посылаются на COM-порт.

Далее, будучи преобразованными в http запрос показания передаются в систему управления.

Полный код программы представлен в приложении 2.

Заключение

Широкая распространенность и доступность современных микроконтроллеров открывает большой простор для их изучения и использования. Возможности контроллеров широки, а относительная простота их освоения позволяет применять их не только на производстве, но и в повседневной жизни для решения бытовых задач. Реализация таких проектов, как «умный дом» или «умная одежда» вполне реальна и уже не является чем-то фантастическим. Что и было продемонстрировано в данной работе. Активное развитие современных разработок подтверждает, что исследование и изучение микропроцессоров и микроконтроллеров актуально и востребовано и позволяет разрабатывать системы различной сложности.

В настоящей работе проанализирована профессиональная литература по теме программирования микроконтроллеров. Выбраны программные и аппаратные средства для создания системы контроля помещения. В ходе работы на базе выбранной среды программирования разработан программный код, позволяющий получать и предоставлять пользователю данные показателей полученные с внешних датчиков. Выполнена сборка действующей модели на основе платформы Arduino и на практических примерах продемонстрирована работа контроллера при обработке заложенных в него программ.

Разработанная в ходе практики система управления успешно внедрена в тестовом режиме на предприятии ООО «ЭЙТИЭМ СОФТВЭАР САППОРТ» и показала достойные результаты. Использование системы позволило оптимизировать условия труда, что сократило время реагирования и повысило трудоспособность сотрудников.

Таким образом, цели работы достигнуты, задачи выполнены.

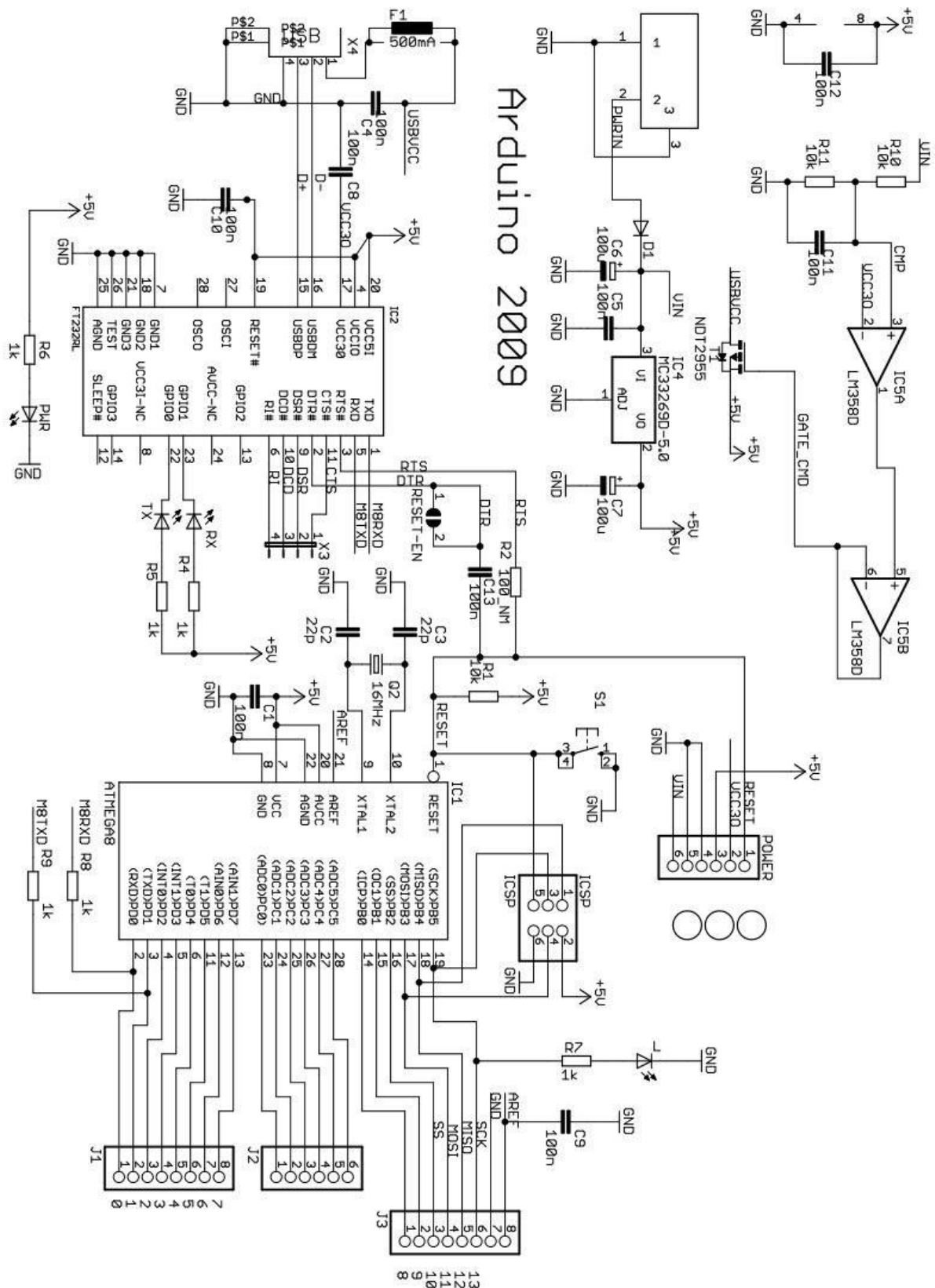
Список литературы

1. Datasheet ATmega168-20PU - Atmel 8- bit Microcontrollers (MCU) 16 Kb Flash 0.5 Kb EEPROM 23 I/O Pins.
2. Баранов В. Н. Применение микроконтроллеров AVR: схемы, алгоритмы, программы, 2-е изд. испр. - М.: Издательский дом "Додэка-XXI", 2014. – 288 с.
3. Белов А. В. Конструирование устройств на микроконтроллерах. - СПб.: Наука и Техника, 2015 – 256 с.
4. Белов А. В. Самоучитель по микропроцессорной технике. - СПб.: Наука и Техника, 2014 – 211 с.
5. Богданов, С. В. Умный дом: монография / С. В. Богданов. - 2-е изд., перераб. и доп. - СПб. : Наука и Техника, 2015. – 208 с.
6. Велт, Т. Дж., Элсенпитер, Р. К. "Умный дом" строим сами / Т. Дж. Велт, Р. К. Элсенпитер. - СПб. : КУДИЦ-Образ, Питер, 2014. – 384 с.
7. Водовозов А. М. Микроконтроллеры для систем Автоматики: Учебное пособие. - Вологда: ВоГТУ, 2014. – 123с.
8. Голубцов М. С. Микроконтроллеры AVR: от простого к сложному / М. С. Голубцов - М.: СОЛОН-Пресс, 2014. – 288 с.
9. Гребнев В. В. Микроконтроллеры семейства AVR фирмы Atmel. - М.: ИП РадиоСофт, 2015 – 176 с.
10. Евстифеев А. В. Микроконтроллеры AVR семейств Tiny и Mega фирмы ATMEL. - М.: Издательский дом "Додэка-XXI", 2014. – 208 с.
11. Монк Саймон. Мейкерство. Arduino и Raspberry Pi. Управление движением, светом и звуком / С. Монк; пер. с англ. М. А. Рейтман. - СПб. : БХВ-Петербург, 2017. – 336 с.
12. Предко М. Руководство по микроконтроллерам. Том 1. / Пер. с англ. под ред. И. И. Шагурина и С. Б. Лужанского – М.: Постмаркет, 2015. – 416 с.

- 13.Предко М. Руководство по микроконтроллерам. Том 2. / Пер. с англ. под ред. И. И. Шагурина и С. Б. Лужанского – М.: Постмаркет, 2015. – 488 с.
- 14.Сопер, М. Э. Практические советы и решения по созданию "Умного дома": самоучитель / М. Э. Сопер; пер. с англ. А. Ю. Карцева. - М. : NT Press, 2014. – 421 с.
- 15.Тесля, Е. «Умный дом» своими руками. Строим интеллектуальную цифровую систему в своей квартире / Е. Тесля. – СПб: Питер, 2008. – 224 с.
- 16.Харке, В. Умный дом. Объединение в сеть бытовой техники и систем коммуникаций в жилищном строительстве: монография / В. Харке; пер. с нем. И. В. Рядченко. - М. : Техносфера, 2014 (Чебоксары). – 287 с. : ил.
- 17.Хартов В. Я. Микроконтроллеры AVR. Практикум для начинающих. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2015. – 240 с.
- 18.Хартов В. Я. Проектирование и отладка программ для микроконтроллеров AVR фирмы Atmel. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 198 с.
- 19.Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си/ 2-е издание - М.: Триумф, 2014. – 610 с.
- 20.Bell C. Beginning Sensor Networks with Arduino and Raspberry Pi - Apress, 2013. — 372 p.
- 21.Cady, Fredrick M. Microcontrollers and microcomputers: principles of software and hardware engineering. – New York – Oxford, Oxford University Press, 2014. – 252 p.
- 22.Dennis A.K. Raspberry Pi Home Automation with Arduino - Packt Publishing, 2013. — 176 p.
- 23.Mike Riley «Programming Your Home Automate with Arduino, Android, and Your Computer» - « The Pragmatic Bookshelf Dallas, Texas • Raleigh, North Carolina », 2014. — 242 p.

24. Steven Goodwin. Smart Home Automation with Linux. Learn how to control your home from your PC / Steven Goodwin. Apress. New York, 2015. 269 p.
25. Официальный сайт фирмы Arduino Software [Электронный ресурс] URL: <http://arduino.cc> (дата обращения: 15.08.2017).
26. Русскоязычный сайт по платформам Arduino [Электронный ресурс] URL: <http://arduino.ru> (дата обращения: 11.06.2017).
27. Официальный сайт фирмы Atmel [Электронный ресурс] URL: <http://www.atmel.com/> (дата обращения: 01.06.2017).
28. Программы для микроконтроллеров [Электронный ресурс] URL: http://cxem.net/software/soft_mcu.php (дата обращения: 15.08.2017).
29. Arduino Uno: Основы программирования [Электронный ресурс] URL: <http://developer.alexanderklimov.ru/arduino/arduino-minimum.php> (дата обращения: 25.07.2017).
30. Цифровая техника в радиосвязи [Электронный ресурс] URL: <http://digteh.ru/> (дата обращения: 15.08.2017).
31. Описание основных функций языка Arduino [Электронный ресурс] URL: <http://freeduino.ru/arduino/lang.html> (дата обращения: 11.06.2017).
32. Программирование Arduino с помощью ArduBlock на примере робота, движущегося по полосе [Электронный ресурс] URL: <http://habrahabr.ru/post/240441/> (дата обращения: 05.09.2017).
33. Официальный сайт платформы MajorDoMo [Электронный ресурс] URL: <https://majordomo.smartliving.ru/> (дата обращения: 18.06.2017).
34. Электронный научный журнал "Программные продукты, системы и алгоритмы" [Электронный ресурс] URL: <http://swsys-web.ru/cloud-computing-basic-concepts-problems.html> (дата обращения: 20.09.2017).
35. Амперка/Вики [Электронный ресурс] URL: <http://wiki.amperka.ru/>
36. Программирование Arduino [Электронный ресурс] URL: http://zelectro.cc/arduino_ide (дата обращения: 11.06.2017).

Принципиальная схема платформы Arduino



Исходный код программы микроконтроллера

```
int cycle_counter = 0;
int old_movement_1 = 0;
int old_temperature1 = 0;
int old_hum = 0;
char buf[80];
#include <dht11.h>
dht11 DHT;
#define DHT11_PIN 4
void sendHTTPRequest() {
    Serial.println(buf);
}
void setup()
{
    Serial.println("Type,\tstatus,\tHumidity (%),\tTemperature (C)");

    pinMode(6, INPUT);
    old_movement_1 = digitalRead(6);
    Serial.begin(115200);
    Serial.println("Start");
}
void loop()
{
    int valid_sensor = 0;
    delay(100);
    cycle_counter++;
    if (cycle_counter > 600) {
```

```

    cycle_counter = 0;
}

int current_movement_1 = digitalRead(6);
if (current_movement_1 != (int)old_movement_1) {
    old_movement_1 = (int)current_movement_1;
    sprintf(buf,                                     "GET
/objects/?object=sensorMovement&op=m&m=statusChanged&status=%i",
(int)current_movement_1);
    sendHTTPRequest();
}
int chk;
int current_temp1 = old_temperature1;
int current_hum = old_hum;

chk = DHT.read(DHT11_PIN);
switch (chk) {
    case DHTLIB_OK:
        break;
    case DHTLIB_ERROR_CHECKSUM:
        Serial.println("Checksum error,\t");
        break;
    case DHTLIB_ERROR_TIMEOUT:
        Serial.println("Time out error,\t");
        break;
    default:
        Serial.println("Unknown error,\t");
        break;
}

```

```

old_temperature1 = (DHT.temperature);
old_hum = (DHT.humidity);
int temp1 = (current_temp1 - (int)current_temp1) * 100;
int hum = (current_hum - (int)current_hum) * 100;
sprintf(buf,
"GET/objects/?object=TempSensor01&op=m&m=tempChanged&t=%0d.%d",
(int)current_temp1, abs(temp1));
sendHTTPRequest();
sprintf(buf,
"GET/objects/?object=humSensor01&op=m&m=humChanged=%0d.%d",
(int)current_hum, abs(hum));
sendHTTPRequest();
delay(5000);

}

```